



# Universidad de Cuenca

## Facultad de Ingeniería

### Escuela de Electrónica y Telecomunicaciones

---

## Diseño e implementación de un adaptador para teléfono analógico de bajo costo

---

---

*TRABAJO DE TITULACIÓN PREVIA A  
LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES*

---

**Autores :**

*Emanuel Alexander Calle Espinoza. CI. 0703820050*

*Gabriel Sebastián Castro Tamay. CI. 1104245566*

**Director :**

*Ing. Darwin Fabián Astudillo Salinas, PhD. CI. 0103907036*

---

Cuenca - Ecuador

2017





# Resumen

La evolución de las [Tecnologías de la Información y Comunicación \(TICs\)](#) ha permitido que el servicio de Internet sea cada vez más asequible, lo que ha permitido que tecnologías como [Voice over IP \(VoIP\)](#) ganen terreno en el mercado. El servicio de telefonía [Internet Protocol \(IP\)](#) tiene un menor costo que la telefonía convencional en cuanto al servicio se refiere, sin embargo, los equipos requeridos para hacer uso de este servicio, como son los teléfonos [IP](#) y dispositivos [Analogue Telephone Adapter \(ATA\)](#), tienen un precio elevado y estos últimos son además muy escasos. Es por esta razón que el presente proyecto plantea diseñar un adaptador para teléfono analógico de bajo costo, que ofrezca a los usuarios una alternativa sencilla y económica para el acceso a [VoIP](#).

El dispositivo que se ha planteado debe ser capaz de brindar el servicio de [VoIP](#) a través de un teléfono convencional. Para esto debe poder acceder a la red, tener implementado un sistema de codificación y decodificación acorde al estándar [VoIP](#), utilizar un protocolo de inicio de sesión para conectarse con un servidor [VoIP](#) y un sistema de detección de marcación para comunicarse con el teléfono.

Debido a que planteamos desarrollar un dispositivo de bajo costo, se utilizó el módulo Wi-Fi ESP8266 para acceder a la red. En este módulo también se implementaron los protocolos de inicio de sesión ([Session Initiation Protocol \(SIP\)](#)) y de tráfico de voz ([Real Time Protocol \(RTP\)](#)). La codificación y decodificación de la voz se realizó mediante el algoritmo A-law, el cual fue implementado en un dSPIC33 de Microchip. Las pruebas de se realizaron usando el servidor de [VoIP](#) Elastix.

**Palabras clave :** [VoIP](#), [ATA](#), [SIP](#), [A-law](#), [RTP](#) .







# Abstract

The evolution of [Information and Communications Technology \(ICT\)](#) has allowed the Internet service to become affordable. This has allowed technologies such as [VoIP](#) to gain a place in the market. The [IP](#) telephony service has a lower cost than conventional telephony when it comes to service, however, the equipment required to make use of this service, such as [IP](#) phones and [ATA](#) devices are expensive and limited. For this reason, this project proposes to design a low cost analogue telephone adapter, that offers users a simple and economic alternative for [VoIP](#) access.

The device that has been put forward must be able to provide [VoIP](#) services through a conventional telephone. In order to achieve this, it must be able to access the network, implement a coding and decoding system according to the [VoIP](#) standard, use a session initiation protocol to connect to a [VoIP](#) server and a dial-up detection system to communicate with the telephone.

Since we proposed to develop an inexpensive device, we used the ESP8266 Wi-Fi module for network access. This module also implements session initiation ([SIP](#)) and voice traffic ([RTP](#)) protocols. Voice coding and decoding was performed using the A-law algorithm, which was implemented in a Microchip dSPIC33. The tests were performed by using the Elastix [VoIP](#) server.

**Keywords :** [VoIP](#), [ATA](#), [SIP](#), [A-law](#), [RTP](#)





# Índice general

Resumen	III
Abstract	V
Índice general	VII
Índice de figuras	IX
Índice de tablas	XI
Dedicatoria	XXI
Agradecimientos	XXIII
Abreviaciones y acrónimos	XXV
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Planteamiento del Problema . . . . .	1
1.3. Objetivos . . . . .	2
1.3.1. Objetivo General . . . . .	2
1.3.2. Objetivos Específicos . . . . .	2
<b>2. VoIP y Protocolos para VoIP</b>	<b>5</b>
2.1. Antecedentes de VoIP . . . . .	5
2.2. VoIP vs Telefonía IP . . . . .	6
2.3. Ventajas de la Telefonía IP . . . . .	6
2.4. Desventajas de la Telefonía IP . . . . .	7
2.5. Calidad de Voz . . . . .	7
2.6. Digitilización de la Voz . . . . .	8
2.6.1. Muestreo . . . . .	8
2.6.2. Cuantificación . . . . .	9



2.6.3. Codificación . . . . .	9
2.7. CODECS (Decodificador y codificador) . . . . .	10
2.8. Protocolos . . . . .	12
2.8.1. Protocolo de Transporte . . . . .	12
2.8.2. Protocolo RTP . . . . .	12
2.8.3. Protocolo SIP . . . . .	13
2.8.3.1. Arquitectura . . . . .	15
2.8.3.2. Intercambio de Mensajes . . . . .	16
<b>3. Diseño del Adaptador</b>	<b>19</b>
3.1. Diseño e implementación del circuito de interfaz . . . . .	19
3.2. Configuración e implementación del módulo Wi-Fi . . . . .	23
3.2.1. Conexión del Adaptador a la Red . . . . .	23
3.2.2. Establecimiento de inicio de sesión . . . . .	23
3.2.3. Establecimiento de invitación de llamada . . . . .	25
3.2.4. Media Session . . . . .	27
3.3. Configuración y operación del microcontrolador . . . . .	28
<b>4. Pruebas y resultados</b>	<b>33</b>
4.1. Señalización telefónica . . . . .	33
4.2. Conexión con la red doméstica . . . . .	35
4.3. Conexión con el servidor VoIP . . . . .	37
<b>5. Conclusiones y Recomendaciones</b>	<b>39</b>
5.1. Conclusiones . . . . .	39
5.2. Recomendaciones . . . . .	40
5.3. Trabajos Futuros . . . . .	40
<b>A. Esquemático</b>	<b>43</b>
<b>Bibliografía</b>	<b>45</b>



# Índice de figuras

1.1. Esquema VoIP . . . . .	2
1.2. Esquema General del Adaptador . . . . .	2
2.1. Esquema Digitalización de la Voz . . . . .	8
2.2. Señal Analógica vs Muestreada . . . . .	9
2.3. Cuantificación de la Señal. <i>Imagen tomada de [1]</i> . . . . .	10
2.4. Codificación de la Señal. <i>Imagen tomada de [1]</i> . . . . .	10
2.5. Comunicación RTP . . . . .	13
2.6. Formato paquete RTP . . . . .	13
2.7. Arquitectura Básica SIP . . . . .	15
2.8. Estructura Genérica de un Mensaje SIP . . . . .	16
2.9. Intercambio Básicos de mensajes SIP . . . . .	17
3.1. Diagrama de bloques del circuito de interfaz . . . . .	20
3.2. Circuito esquemático de la primera parte del bloque 1 . . . . .	20
3.3. Circuito Esquemático de la segunda parte del bloque 1 . . . . .	21
3.4. Circuito esquemático del bloque 2 . . . . .	21
3.5. Marcación por pulsos . . . . .	22
3.6. Circuito esquemático del timbre . . . . .	22
3.7. Flujograma para conexión a red . . . . .	24
3.8. Formato básico de mensaje SIP . . . . .	25
3.9. Formato básico de mensaje SIP . . . . .	26
3.10. Flujos de llamada . . . . .	28
3.11. Distribución de pines del microcontrolador . . . . .	28
3.12. Diagrama de estados del dispositivo . . . . .	29
4.1. Señal de estado de línea . . . . .	33
4.2. Señal de marcación por pulsos . . . . .	34
4.3. Señales informativas . . . . .	34
4.4. Página de Configuración . . . . .	35



4.5. Botón Guardar . . . . .	36
4.6. Botón Escanear . . . . .	36
4.7. Conexión a la red . . . . .	37
4.8. Servidor Elastix . . . . .	37
4.9. Estado de extensiones . . . . .	38



# Índice de tablas

2.1. Codecs . . . . .	11
2.2. Características SIP . . . . .	14
3.1. Tiempos de cadencia utilizados en Ecuador . . . . .	21
3.2. Formato SDP . . . . .	27
3.3. Cálculo Longitud de Contenido . . . . .	27
3.4. Codificación binaria A-law . . . . .	30
3.5. Decodificación binaria A-law . . . . .	31







Yo, Emanuel Alexander Calle Espinoza, autor del trabajo de titulación “Diseño e implementación de un adaptador para teléfono analógico de bajo costo”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 25 de Octubre de 2017

---

Emanuel Alexander Calle Espinoza  
070382005-0





Yo, Emanuel Alexander Calle Espinoza, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Diseño e implementación de un adaptador para teléfono analógico de bajo costo”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 29 de Agosto de 2017

---

Emanuel Alexander Calle Espinoza  
070382005-0





Yo, Gabriel Sebastián Castro Tamay, autor del trabajo de titulación “Diseño e implementación de un adaptador para teléfono analógico de bajo costo”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 25 de Octubre de 2015

---

Gabriel Sebastián Castro Tamay

110424556-6





Yo, Gabriel Sebastián Castro Tamay, en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación “Diseño e implementación de un adaptador para teléfono analógico de bajo costo”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 29 de Agosto de 2017

---

Gabriel Sebastián Castro Tamay

110424556-6







# Dedicatoria

Le dedico este logro de mi vida a los seres que más amo, mis padres Nube y Oscar que siempre me han apoyado en este difícil camino. A mi abuelita, mi hermana y a toda mi familia que de alguna forma u otra siempre estuvieron a mi lado.

**Alexander Calle**

A mis padres Reinaldo y Bella por estar siempre presentes, por su constante amor y apoyo. A mi hermana Anahí, a quien quiero mucho. A Pablo, mi hermano y amigo, quien en gran medida me ayudó a convertirme en la persona que soy. Y a mis amigos que de una u otra forma estuvieron ahí.

**Gabriel Castro**





# Agradecimientos

En primer lugar, agradecemos a nuestro director de tesis, Ing. Fabián Astudillo, que nos supo guiar y apoyar durante la realización de este proyecto.

A nuestros padres, por el apoyo incondicional y por siempre haber creído en nosotros.

A toda nuestros amigos que estuvieron a nuestro lado durante el transcurso de esta aventura.

Muchas Gracias.

**Los Autores**





# Abreviaciones y Acrónimos

- A/D** Analógico/Digital. [10](#)
- ACK** *Acknowledgement*. [16](#)
- ADPCM** *Adaptive Differential Pulse-Code Modulation*. [11](#)
- AP** *Acces Point*. [35](#)
- ARPANET** *Advanced Research Projects Agency Network*. [5](#)
- ASCII** *American Standard Code for Information Interchange*. [14](#)
- ATA** *Analogue Telephone Adapter*. [1](#), [2](#), [5](#), [6](#), [15](#), [23](#)
- AVP** *Audio Video Profile*. [27](#)
- 
- CPU** *Unidad Central de Procesamiento*. [10](#)
- CS-ACELP** *Conjugate-Structure Algebraic Code Excited Linear Prediction*. [11](#)
- CSRC** *Contributing Source Identifier*. [13](#)
- 
- DAC** *Digital Analogue Converter*. [20](#), [29](#)
- DSP** *Digital Signal Processor*. [6](#)
- DTMF** *Dual-Tone Multi-Frecuency*. [5](#), [39](#)
- 
- EEPROM** *Electrically Erasable Programmable Read-Only Memory*. [23](#), [35](#)
- 
- FTP** *File Transfer Protocol*. [12](#), [23](#)
- 
- HTML** *HyperText Markup Language*. [23](#)
- HTTP** *Hypertext Transfer Protocol*. [12](#), [14](#), [15](#), [23](#)
- 
- IEEE** *Institute of Electrical and Electronics Engineers*. [1](#)
- IETF** *Internet Engineering Task Force*. [6](#), [14](#), [23](#)
- IP** *Internet Protocol*. [1](#), [5–8](#), [10](#), [11](#), [14](#), [15](#), [23](#), [27](#), [35](#)
- 
- LAN** *Local Area Network*. [5](#)
- LD-CELP** *Low Delay Code Excited Linear Prediction*. [11](#)
- 
- MCU** *Microcontroller Unit*. [1](#), [19](#)
- MD5** *Message-Digest Algorithm 5*. [25](#)



- MIME** *Multipurpose Internet Mail Extensions*. [15](#)
- MIPS** *Millones de Instrucciones por Segundo*. [10](#)
- MP-MLQ** *Multi-Pulse Maximum Likelihood Quantization*. [11](#)
- PCM** *Pulse Code Modulation*. [11](#), [30](#)
- PCMA** *Pulse Code Modulation A-law*. [26](#), [27](#)
- PIC** *Programmable Interrupt Controller*. [28](#)
- QoS** *Quality of Service*. [5](#), [12](#)
- RC** *Resistor-Capacitor*. [22](#)
- RFC** *Request for Comments*. [12–14](#), [23](#), [40](#)
- RTCP** *Real Time Control Protocol*. [12](#)
- RTP** *Real Time Protocol*. [12](#), [13](#), [19](#), [23](#), [26](#), [27](#)
- SB-ADPCM** *Sub Band Adaptive MP-MLQ Low Delay Code Excited Linear Prediction*. [11](#)
- SDP** *Session Description Protocol*. [26](#), [27](#)
- SIP** *Session Initiation Protocol*. [1](#), [3](#), [6](#), [14–17](#), [23](#), [25](#), [39](#)
- SIPS** *Secure SIP*. [15](#)
- SMTP** *Simple Mail Transfer Protocol*. [12](#)
- SSID** *Service Set Identifier*. [23](#), [35](#)
- SSRC** *Synchronization Source*. [13](#)
- TCP** *Transmission Control Protocol*. [12](#), [14](#), [23](#)
- TICs** *Tecnologías de la Información y Comunicación*. [1](#)
- UA** *User Agent*. [15](#), [37](#)
- UAC** *User Agent Client*. [15](#)
- UAS** *User Agent Server*. [15](#)
- UDP** *User Datagram Protocol*. [12](#), [14](#), [19](#), [23](#)
- UIT** *Unión Internacional de Telecomunicaciones*. [10](#)
- UIT-T** *Grupo de Normalización de las Telecomunicaciones de la UIT*. [5](#), [8](#), [21](#)
- URI** *Uniform Resource Identifier*. [15](#)
- URL** *Uniform Resources Locator*. [25](#)
- VoIP** *Voice over IP*. [1–3](#), [5–7](#), [14](#), [23](#), [37](#)
- WAN** *Wide Area Network*. [5](#)
- WEB** *World Wide Web*. [23](#), [35](#)



# Capítulo 1

## Introducción

### 1.1. Antecedentes

En la actualidad, existe una evidente evolución de las **TICs**, donde el Internet de alta velocidad ha permitido que las comunicaciones analógicas tiendan a migrar a sus contrapartes digitales para aprovechar la gran capacidad que ofrece. Un claro ejemplo de este fenómeno es el servicio de **VoIP** [2], el cual ha sido adoptado por muchos proveedores de telefonía móvil y fija.

Con el cambio a **VoIP**, la problemática que enfrentarían los usuarios es la adquisición de un equipo, ya sea un teléfono **IP** o un adaptador **ATA** [3]. En el mercado nacional los teléfonos **IP** tienen un costo elevado, mientras que los equipos **ATA** son escasos en el mercado y también tienen un costo alto.

### 1.2. Planteamiento del Problema

En este proyecto se plantea implementar un adaptador **ATA** de bajo costo, ofreciendo así una alternativa económica para hacer este tipo de tecnología más asequible a los usuarios de telefonía fija. Una descripción general del funcionamiento del dispositivo se puede apreciar en la Figura 1.1 Donde el teléfono analógico **A** está conectado al adaptador **ATA**. Este adaptador **ATA** estará conectado al enrutador mediante Wi-Fi y establecerá sesión con el servidor mediante el protocolo **SIP** [4]. Además se realizará la codificación de la voz mediante el estándar G.711 [5]. Por último, el servidor establecerá la comunicación desde teléfono **A** al equipo **B**.

De forma más detalla el dispositivo **ATA** cuenta con las etapas mostradas en la Figura 1.2. El **circuito de interfaz** realiza el acoplamiento de la señal proveniente del teléfono para poder ser procesadas por el **MCU** y viceversa. El **MCU** se encarga de realizar la adquisición de la señal de voz analógica para su posterior codificación y decodificación. En el **módulo Wi-Fi** se realiza la ejecución de los diferentes métodos dentro del protocolo **SIP**. Este también se encarga del procesamiento de los paquetes de información según el estándar **IEEE 802.11 b/g/n/e/i** [6].

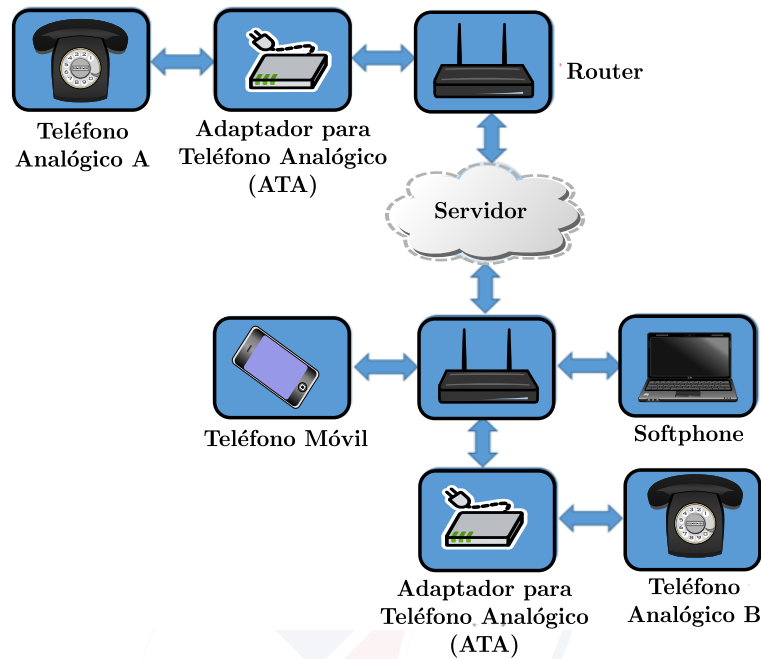


Figura 1.1: Esquema VoIP

Y por último este módulo empieza a enviar y recibir paquetes de voz a través del servidor.

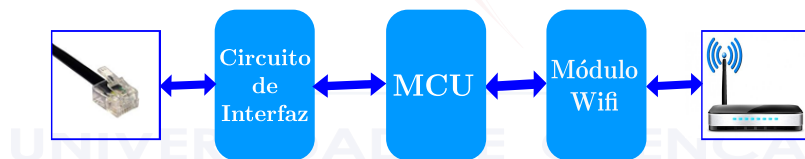


Figura 1.2: Esquema General del Adaptador

### 1.3. Objetivos

#### 1.3.1. Objetivo General

Diseñar e implementar un dispositivo **ATA** que brinde el servicio de **VoIP** sobre equipos de telefonía fija.

#### 1.3.2. Objetivos Específicos

1. Implementar la codificación y decodificación de señales de voz mediante el estándar G.711.
2. Elaborar una interfaz que adapte las señales provenientes del teléfono analógico para que puedan ser procesadas digitalmente.





3. Desarrollar un método de detección de marcación.
4. Diseñar e implementar una solución para el establecimiento de llamadas con [VoIP](#) y configuración de sesión [SIP](#).



UNIVERSIDAD DE CUENCA  
*desde 1867*



---

UNIVERSIDAD DE CUENCA  
*desde 1867*

---



## Capítulo 2

# VoIP y Protocolos para VoIP

### 2.1. Antecedentes de VoIP

Con el desarrollo de *Advanced Research Projects Agency Network (ARPANET)*, la transmisión de voz sobre IP aparece por primera vez como un desarrollo experimental para comunicar dos PCs por voz dentro la pequeña red. Se desarrollaron varias técnicas de digitalización de voz, mecanismos de control y priorización de tráfico de paquetes, protocolos de transmisión de datos en tiempo real, así como el desarrollo de nuevos estándares para el control de servicio sobre redes IP (*Quality of Service (QoS)*). VoIP se creó inicialmente para reducir el ancho de banda, tomando ventaja de los procesos de compresión de voz que se diseñaron para los sistemas celulares. Luego tuvo aplicación en la red de servicios integrados LAN e Internet; y posteriormente, se migró a la red WAN, la cual denominaron IP-Telephony.

VoIP es una tecnología que permite la transmisión de voz a través de las redes IP, surgió como resultado del trabajo de un grupo de estudiantes en Israel. Posteriormente, Volcatel lanzó el primer *softphone* que llamaron “Internet Phone Software”. Unos años más tarde, MCI desarrolló el proyecto llamado VAULT, una nueva arquitectura de red que permite interconectar y combinar las redes tradicionales de telefonía con redes de datos.

Debido a la existencia del estándar H.323 de la *Grupo de Normalización de las Telecomunicaciones de la UIT (UIT-T)*, que abarcaba la mayor parte de las necesidades para la integración de la voz, se decidió que éste sea fuera la base para VoIP. VoIP tiene como objetivo la interoperabilidad entre los diferentes fabricantes, estableciendo aspectos como: supresión de silencios, codificación de la voz y direccionamiento, servicios de directorios y la transmisión de señalización por tonos multifrecuencia *Dual-Tone Multi-Frequency (DTMF)*. En los siguientes años se comienzan a fabricar los primeros ATA/Gateways que permitían la comunicación entre un teléfono convencional a una PC y finalmente se dieron las primeras comunicaciones entre dos teléfonos convencionales. Las primeras plataformas corporativas utilizando VoIP se comienzan comercializar por Cisco en el año de 1999. En ese mismo año 3Com aprovechó con éxito las



capacidades de VoIP, desarrollando un sistema basado en *Digital Signal Processor (DSP)*, que fue contruido en parte sobre un servidor de Microsoft Windows NT. VoIP representaba más del 3 % del tráfico de voz. Es cuando Mark Spencer, estudiante de la universidad de Auburn crea Asterisk. El protocolo de señalización SIP, desarrollado por la *Internet Engineering Task Force (IETF)*, empieza a relegar al protocolo H.323. Jan Friis y Niklas Zennstrom, dos estudiantes universitarios, crean Skype. Linksys, una división de Cisco lanza un teléfono móvil IP llamado iPhone que cuenta con clientes Skype y Yahoo! Messenger [7].

## 2.2. VoIP vs Telefonía IP

No es lo mismo VoIP que telefonía IP, estos dos términos están muy estrechamente relacionados y se pueden apreciar ciertas diferencias. VoIP es una tecnología que permite la transmisión de la voz a través de redes IP en forma de paquetes. La telefonía IP es una aplicación inmediata de la tecnología VoIP, de tal forma que permita la realización de llamadas telefónicas convencionales sobre redes IP, en lugar de ser transportados vía red telefónica convencional, utilizando teléfonos convencionales y ATA o Gateways. La telefonía IP sigue varios pasos para poder realizar una llamada a través de Internet los cuales son:

1. Conversión de la señal de voz analógica a digital
2. Compresión de la señal y empaquetado IP
3. Transmisión de los paquetes de voz

Durante la recepción se realiza el proceso inverso [6]. Existen 3 tipos de llamadas: *softPhone* a *softPhone*, *softPhone* a teléfono y de teléfono a teléfono.

## 2.3. Ventajas de la Telefonía IP

El costo de la comunicación es la principal ventaja, una llamada utilizando telefonía convencional requiere de una red de centralitas telefónicas conectadas entre sí mediante fibra óptica y satélites de comunicación, además de los cables de cobre que unen los teléfonos con la centralitas. Las inversiones necesarias para crear y mantener estas infraestructuras hacen que las llamadas tengan un costo muy elevado. En la telefonía IP, se comprimen los datos de la señal de voz y se utiliza una red de paquetes cuando sea necesario. Los paquetes de diferentes llamadas e incluso de diferentes tipos de datos, pueden viajar sobre la misma línea al mismo tiempo. Además, el acceso a Internet cada vez es más barato, con lo cual el costo de la llamada puede llegar a ser muy bajo o incluso no tener costo alguno. Gran parte de los proveedores de VoIP que proporcionan servicios a las operadoras de telefonía convencional cobran tarifas separadas. Varios de los servicios son:

1. Identificación de llamadas
2. Servicios de llamadas en espera



3. Servicio de transferencia de llamadas
4. Repetición de llamadas
5. Devolver llamadas
6. Llamadas de 3 líneas
7. Desviar llamadas
8. Enviar llamadas directamente al buzón de voz
9. Dar a una llamada tono de ocupado.

Con el avance tecnológico y el desarrollo de códecs para VoIP, hoy en día se pueden establecer comunicaciones a través de anchos de banda reducidos, ya que la compresión de los datos mediante los códecs es mayor, generando menos tráfico. Con respecto a los equipos, estos se pueden utilizar en cualquier parte del planeta, esto quiere decir que un usuario puede realizar una llamada fuera de su país de origen, siempre y cuando éste pueda establecer una conexión a Internet [6].

## 2.4. Desventajas de la Telefonía IP

Durante una llamada por VoIP se puede producir ciertos retrasos en la llegada de los paquetes e incluso pérdidas de información, pero esto se debe a las limitaciones del servicio de Internet que se haya contratado. Se puede producir deterioros dentro de la comunicación a través de VoIP. Esto por lo general sucede cuando existen congestiones de red. VoIP requiere de una conexión eléctrica continua. En caso de haber un corte eléctrico, los teléfonos IP dejarían de funcionar, en cambio, en los teléfonos convencionales esto no sucede a menos que se trate de un teléfono inalámbrico. En VoIP existe un problema con las llamadas de emergencia, debido a que la telefonía IP utiliza direcciones IP para poder identificar un número telefónico, y esta no puede ser asociada con una ubicación geográfica [6].

## 2.5. Calidad de Voz

La calidad de voz es la capacidad de la red para satisfacer las demandas de tráfico en tiempo real en términos de diferentes parámetros: ancho de banda, pérdidas, retardo, *jitter* y eco.

**Jitter:** En redes de paquetes no es posible garantizar que todos los paquetes sigan una misma ruta, al contrario de lo que ocurre en redes de conmutación. Como consecuencia, cada paquete llegará a su destino atravesando un número distintos de nodos, por lo que alcanzarán su objetivo con un retardo diferente. Esta variabilidad de retardo se lo conoce como *jitter*. Para disminuir este retardo se utilizan *buffers* de supresión de *jitter*. La supresión consiste en el almacenamiento de los paquetes durante un determinado tiempo para que estos puedan reordenarse y reproducirse en el orden correcto [2].

**Pérdidas:** Las pérdidas de paquetes son resultado del descarte de paquetes que se producen en los nodos de la red como consecuencia de congestiones. El efecto de estas pérdidas se ven reflejadas en una disminución en la calidad de voz, puesto que faltan paquetes para su correcta reconstrucción. Una solución factible sería mejorar la arquitectura de la red, sustituyendo líneas y routers por otros de mayor capacidad. Sin embargo, si se produce un aumento en el tráfico el problema volverá a aparecer [2].

**Retardo o Latencia:** El retardo o latencia es el tiempo que le toma a la señal de voz viajar desde su origen al destino. El retardo máximo aceptable marca un umbral entre los 150 ms a 200 ms según la recomendación G.144 de la UIT-T [2, 8].

**Eco:** El eco es un fenómeno muy común en las redes telefónicas convencionales y de paquetes. Se produce cuando el usuario escucha parte de su propia voz junto con la del otro usuario o en ausencia de este. Las causas del eco son variadas. Primero se encuentra el eco acústico, que se debe a un acoplamiento entre el micrófono y el auricular del teléfono. Otro tipo de eco es el eco eléctrico, este se produce por un desacople de impedancias en el receptor [2].

## 2.6. Digitilización de la Voz

Para poder realizar la transmisión de voz sobre IP es necesario realizar una digitalización de la voz, para esto se realiza el proceso mostrado en la Figura 2.1.

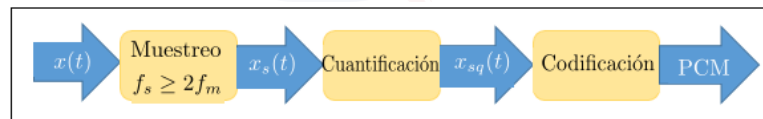


Figura 2.1: Esquema Digitalización de la Voz

### 2.6.1. Muestreo

El muestreo de una señal consiste en seleccionar valores de una señal analógica a lo largo del tiempo a una frecuencia determinada. Esta frecuencia debe seleccionarse con cuidado para evitar pérdida de información o submuestreo. En la Figura 2.2 se observa la señal analógica y la muestreada.

Al digitalizar una señal, se debe cumplir con el Teorema de Nyquist. Este teorema establece que una señal  $x(t)$  cuya frecuencia es  $f_m$ , tiene que ser muestreada a una frecuencia  $f_s$  de por lo menos el doble de la frecuencia  $f_m$ ,  $f_s \geq 2f_m$ . La tasa de muestreo  $s_m = 2f_m$  es conocida como la tasa de Nyquist. Para frecuencias de muestreo menores que la frecuencia de Nyquist, resulta complejo recuperar la señal original a partir de sus muestras debido a que se produce una dispersión energética que distorsiona la señal resultante. Sin embargo, para valores mayores esto no ocurre [2].

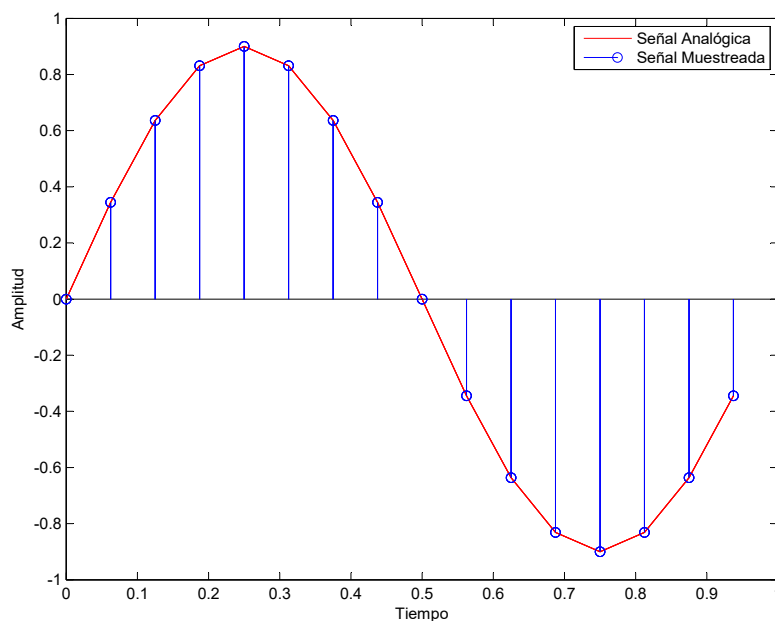


Figura 2.2: Señal Analógica vs Muestreada

### 2.6.2. Cuantificación

El resultado del muestreo es un conjunto de valores tomados de la señal, pero la señal sigue siendo continua en amplitud y es necesario discretizarla también en este dominio: asignar a cada una de la muestras un valor de uno de los  $M$  posibles y mantener este valor hasta la siguiente muestra. Este proceso se lo conoce como cuantificación.

El objetivo de la cuantificación es conseguir que la señal pueda representarse en número finito de  $N$ -bits, con el que se podrán representar hasta  $2^N - 1$  valores diferentes. El resultado será la representación digital de la señal [2], la cual se muestra en Figura 2.3.

### 2.6.3. Codificación

La codificación consiste en asignar un código binario a cada uno de los valores discretos de la señal. Con  $k$  bits se puede codificar  $M$  valores, siendo  $M = 2^k$ . Un ejemplo se puede ver en la Figura 2.4.

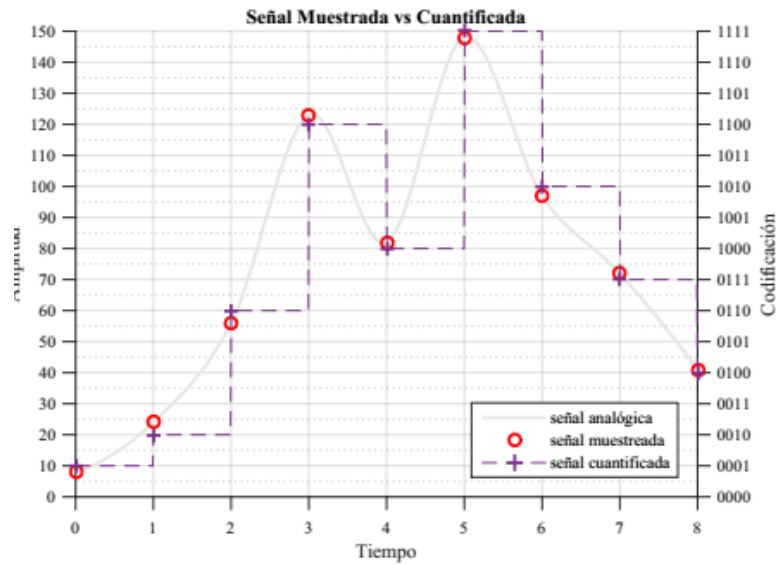


Figura 2.3: Cuantificación de la Señal. *Imagen tomada de [1]*

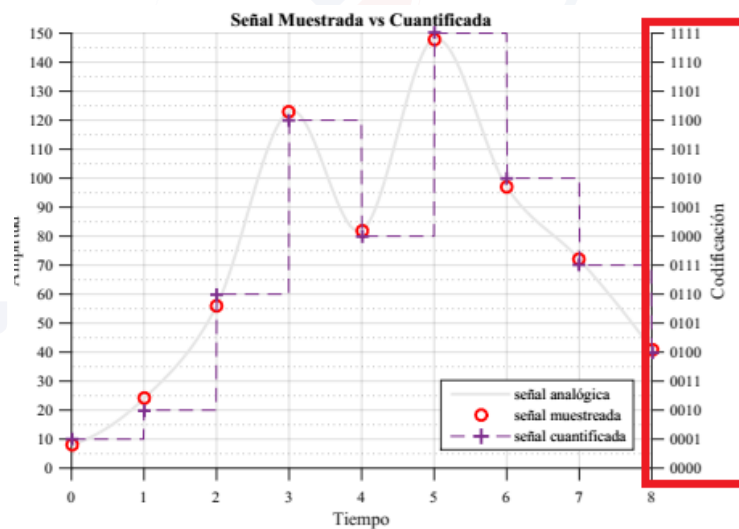


Figura 2.4: Codificación de la Señal. *Imagen tomada de [1]*

## 2.7. CODECS (Decodificador y codificador)

Además de llevar a cabo la conversión A/D, es necesario comprimir la secuencia de datos que supone un ahorro en el ancho de banda. La Tabla 2.1 muestra algunos de los estándares de codificación más importantes cubiertos por la UIT, como se puede observar, la calidad es menor cuanto mayor es la compresión, además que requiere mayor consumo de CPU (MIPS).

La salida del códec es una secuencia de *bytes*, que se convierten en paquetes IP y se trans-





portan a través de la red [IP](#) hasta su destino[9].



Codec	BW(khz)	Frec. de muestreo(khz)	Tramas(ms)	MOS
G.711( <a href="#">PCM</a> )	64	8	0,125	4,4
G.721( <a href="#">ADPCM</a> )	32	8	0,125	4,2
G.722( <a href="#">SB-ADPCM</a> )	48/56/64	16	0,250	
G.728( <a href="#">LD-CELP</a> )	16	8	0,625	4,1
G.729( <a href="#">CS-ACELP</a> )	8	8	10	4,1
G.723.1( <a href="#">MP-MLQ</a> )	6,3	8	30	3,9

Tabla 2.1: Codecs

## 2.8. Protocolos

### 2.8.1. Protocolo de Transporte

Los servicios de este protocolo consisten en el transporte de *bits* entre dos aplicaciones de red. El nivel de transporte en Internet ofrece dos tipos de servicios, según cual sea el protocolo que se utilice:

- Si se requiere la entrega garantizada de los datos al destinatario, sin errores, pérdidas, ni datos duplicados, se utilizará un servicio orientado a la conexión **TCP**. Ejemplos de estas aplicaciones son: **FTP**, **SMTP** Y **HTTP**. El protocolo **TCP** lleva a cabo un control de errores y de flujo de extremo a extremo.
- Por el contrario, si no se requiere una entrega garantizada de los paquetes y no importa que se pierdan o dupliquen los *bits*, se emplea un servicio no orientado a la conexión **UDP**. El protocolo **UDP** se usa en aplicaciones que no requieran una alta fiabilidad con el fin de reducir los retardos asociados a **TCP**.

### 2.8.2. Protocolo RTP

Ya que el protocolo **TCP** no puede realizar un envío de datos en tiempo real y el protocolo **UDP** no conlleva a una confirmación de datos sobre el contenido, ya que en estos dos protocolos no existe un concepto apropiado de **QoS** para la transmisión de voz, surgió el protocolo **RTP**.

El principal objetivo de este protocolo es el de brindar un transporte de datos de extremo a extremo para aplicaciones que requieran transmisión de datos en tiempo real en redes *unicast* o *multicast*: Videoconferencias, difusión de audio/video, simulaciones, adquisición de datos.

El protocolo **RTP** se encarga solo del transporte de datos, mientras que el protocolo encargado del control de los datos es el *Real Time Control Protocol* (**RTCP**). El protocolo **RTP** proporciona servicios de una red extremo a extremo para la transmisión de datos en tiempo real (Figura 2.5), este es un protocolo independiente de transporte y de red aunque en mayor parte se utiliza sobre el protocolo **UDP**[2, 10].

El formato del paquete está estructurado como se muestra en la Figura 2.6.

Donde:

- V:** Es el número de versión **RTP**, contiene dos *bits* y la versión definida por la especificación **RFC 3550** especifica que  $v = 2$ .
- P (padding):** Si el *bit* se encuentra en 1, significa que existe uno o más *bytes* al final del paquete que no corresponden a la carga útil. El *byte* que se encuentra al final del paquete indica el número de *bytes* de relleno.
- X:** Si el *bit* de extensión está colocado, entonces el encabezado fijo es seguido por una extensión del encabezado. Este mecanismo de la extensión posibilita implementaciones para añadir información al encabezado **RTP**.

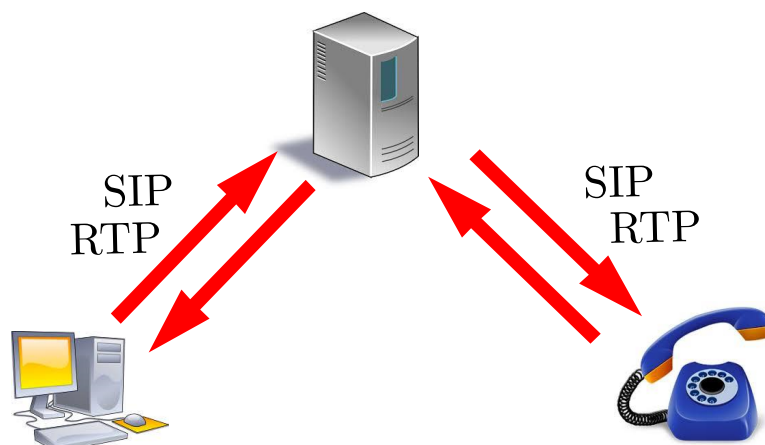


Figura 2.5: Comunicación RTP

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=2		P	X	CC				M	PT							Número de Secuencia															
Estampa de Tiempo																															
Indicador de Fuente de sincronizacion (SSRC)																															
Identificador de Fuente de contribuyente (CSRC)																															
...																															
Carga Útil																															
...																															

Figura 2.6: Formato paquete RTP

**CC o conteo *Contributing Source Identifier (CSRC)*:** Representa al número de identificadores **CSRC** que sigue el encabezado fijo. Esto es por si existe 2 o más usuarios **RTP**.

**M o marcador:** Un *bit* de marcador definido por el perfil.

**PT o payload:** Describe el formato de carga útil que se encuentran en una tabla de perfiles de medios dentro del [RFC 1890](#).

**Número de secuencia:** Es un número único de paquete que identifica la posición de este en la secuencia de paquetes.

**Timestamp:** Refleja el instante de muestreo del primer *byte* en la carga útil.

***Synchronization Source (SSRC)*:** Identifica la fuente de sincronización. Si el valor de **CSRC** es cero, entonces la fuente de carga útil es la fuente de sincronización.

****CSRC:**** Identifica las fuentes contribuyentes para la carga útil. El número de fuentes está indicado por el campo conteo **CSRC**.

### 2.8.3. Protocolo SIP

Este protocolo de señalización en redes de voz sobre paquetes realizan muchas funciones similares a las que llevan a cabo sus homólogos en redes telefónicas conmutadas (establecimiento

de sesión, señales de progreso de llamadas, gestión de usuarios en la llamada, etc.).

Aplicado a la voz sobre paquetes, tiene como objetivo mantener la interfaz con el usuario, es decir, genera tonos y señales necesarios para que el usuario no perciba que la tecnología de soporte de llamadas telefónicas ha cambiado.

En 1999 el [IETF](#) publicó el protocolo [SIP](#) como alternativa para el Protocolo H.323 bajo la [RFC](#) 2543. Se trata de un protocolo de control de la capa de aplicación que define como establecer, modificar o finalizar una sesión entre dos o más extremos.

	<a href="#">SIP</a>
Organismo de Estandarización	<a href="#">IETF</a>
Arquitectura	Distribuida
Versión Actual	<a href="#">RFC</a> 3261
Responsable de Control de Llamadas	Servidor <i>Proxy</i>
Puntos Finales	Agente de Usuario
Transporte	<a href="#">TCP</a> o <a href="#">UDP</a>
Soporte Multimedia	SI
DTMF-relay	<a href="#">RFC</a> 2833
Fax-relay	T.38
Servicios Suplementarios	Proporcionado por el Servidor <a href="#">VoIP</a>

Tabla 2.2: Características SIP

El sistema de mensajes que intercambia [SIP](#) está basado en [HTTP](#), y se emplea en procesos de registro, y para establecer entre que direcciones [IP](#) y puertos [TCP/UDP](#) intercambiarán datos los usuarios. En este sentido, su sencillez hace que este protocolo sea ideal para el desarrollo de nuevos modelos y herramientas de comunicación, asimismo como la telefonía y videoconferencia [IP](#). Aparte de los usuarios, en la arquitectura [SIP](#) se encuentran los servidores *Proxy*, de registro, de redirección y de ubicación, que son los elementos claves para ofrecer “presencia” y “movilidad”. Durante el proceso de registro, un usuario envía un mensaje [SIP](#) para indicar su dirección [IP](#) (dirección física) o [URL](#) (dirección lógica) al servidor de **registro** para que le preste este servicio. A su vez, el servidor de **registro** informa de la localización del usuario a un servidor jerárquicamente superior conocido como de **ubicación**. Cuando un usuario [SIP](#) desea establecer una comunicación con otro, envía un mensaje [SIP](#) a su servidor *Proxy* para conocer la dirección física que tiene el destinatario en ese momento. En este instante, el usuario que realizó la llamada indicará al servidor *Proxy* la dirección lógica del usuario receptor de la llamada, por ejemplo, la dirección de correo electrónico, y con esto, el servidor *Proxy* consultará al servidor de **ubicación**, el cual resolverá la dirección física del usuario receptor de la llamada. En cuanto al tipo de sesión, el protocolo [SIP](#) no establece ninguna limitación y ofrece enormes posibilidades. También se basa en el modelo de Internet y usa códigos [ASCII](#) al igual que [HTTP](#). Su direccionamiento es igual al del correo electrónico: userID@host, donde userID

puede ser el número de teléfono o simplemente un nombre [4].

En el cuerpo del mensaje se tiene los siguientes campos:

**Via:** Cada dispositivo SIP que origina o recibe un mensaje SIP marca su propia dirección en este campo, usualmente escrito como un *host* que se puede resolver en una dirección IP.

**Branch:** Es el identificador de la transacción, relacionando la respuesta con la petición.

**Max-Forward:** Se inicializa en un número entero grande y se va decrementando por cada servidor SIP, que recibe y envía la petición.

**Call-ID:** Es usado para mantener la pista de una sesión SIP en particular.

**CSeq:** Es un mecanismo para identificar y ordenar las transacciones.

**From:** Indica la identidad lógica del dispositivo que genera la petición.

**Tag:** Es un mecanismos de identificación de dialogo.

**To:** Especifica el destinatario lógico de la petición o la dirección de registro del usuario.

**Contact:** Provee un *Uniform Resource Identifier* (URI) SIP o *Secure SIP* (SIPS) que puede ser usado para especificar la instancia del *User Agent* (UA) para solicitudes posteriores.

**Allow:** Indica que servicios desea establecer en la conexión.

**Content-Length:** Indica si existen mensajes adicionales como cifrado *Multipurpose Internet Mail Extensions* (MIME).

### 2.8.3.1. Arquitectura

La arquitectura de SIP es muy similar a la de HTTP: las solicitudes del cliente son enviadas a un servidor, este las procesa y envía una respuesta al cliente.

Los agentes de usuario (UA) son los terminales de los que parten las solicitudes de iniciar o terminar una llamada. Los UA se dividen en dos entidades: el agente de usuario cliente (*User Agent Client* (UAC)), que se encarga de iniciar las sesiones SIP, y el agente de usuario servidor (*User Agent Server* (UAS)) responsable de aceptar las peticiones de establecimiento de sesiones recibidas. Ambas partes pueden terminar una sesión en curso. Los agentes de usuarios pueden ser teléfonos IP, *softphone* o *gateways/ATA* [2, 11].



Figura 2.7: Arquitectura Básica SIP

### 2.8.3.2. Intercambio de Mensajes

El diálogo entre los cliente y los servidores SIP se basa en el intercambio de mensajes de texto. La estructura genérica de un mensaje SIP, ya sea de petición o respuesta, es muy sencilla: línea de comienzo, uno o más campos como se observa en la Figura 2.8, una línea vacía que indica el final de los campos y el cuerpo del mensaje.

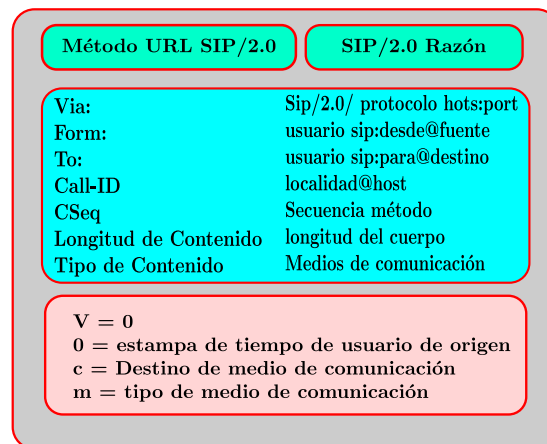


Figura 2.8: Estructura Genérica de un Mensaje SIP

Los mensajes de petición son enviados por las entidades cliente a las entidades servidor. Generalmente, toda petición tiene asociado una respuesta del servidor, excepto el ACK que no requiere una respuesta. Todas las repuestas SIP tienen asociado un código numérico, que indica el resultado del intento de servir la petición al cliente[2, 11].

Las peticiones SIP son:

INVITE: Mensaje inicial de invitación enviado por el usuario que llama.

ACK: Respuesta del agente de llamadas ante el mensaje de aceptación de la llamada.

BYE: Señal de terminación de una llamada.

CANCEL: Cancela una petición.

REGISTER: Empleado por los usuarios para registrar su dirección de contacto.

OPTIONS: Consulta a un agente de usuarios acerca de sus capacidades.

INFO: Contiene información.

Las respuestas por código numérico son:

1xx: Mensajes de información

2xx: Éxito

3xx: Mensaje de desvío

4xx: Error en la petición

5xx: Error en el servidor

**6xx:** Error general

Los seis pasos básicos implicados en el manejo de sesiones SIP se observan en la Figura 2.9 y son:

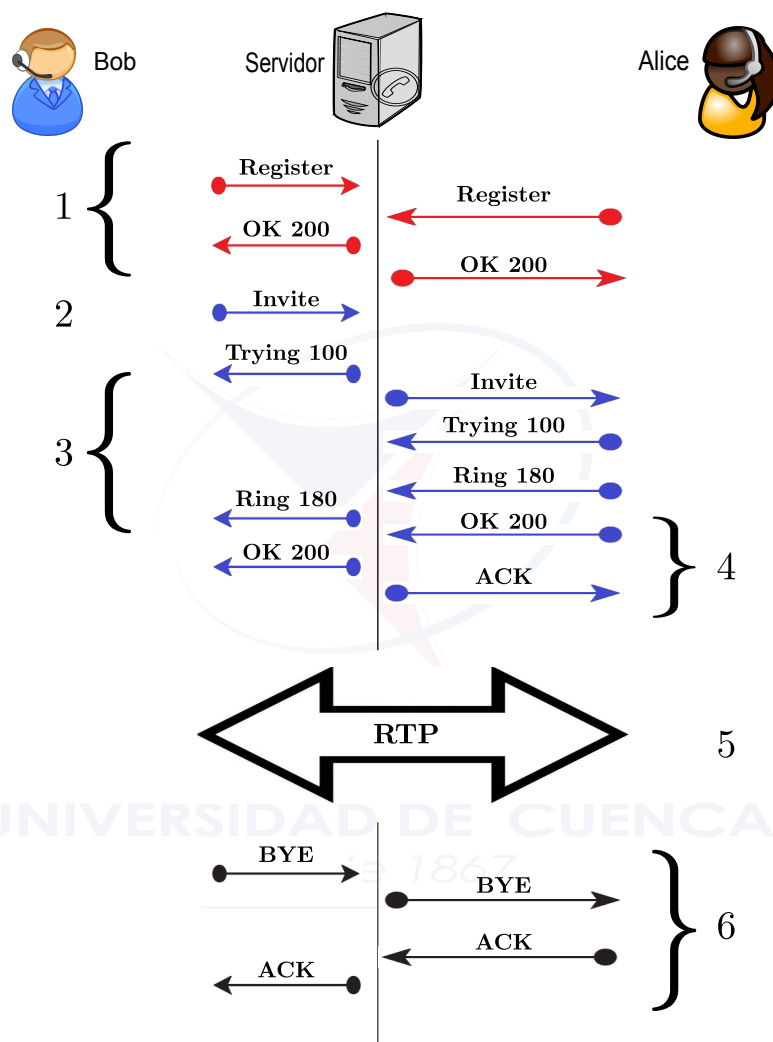


Figura 2.9: Intercambio Básicos de mensajes SIP

1. Registro, iniciación y localización del usuario
2. Descripción de la sesión multimedia que se pretende establecer
3. Aceptación de la petición de conexión del otro extremos
4. Establecimiento de la llamada
5. Comunicación
6. Terminación de la llamada



---

UNIVERSIDAD DE CUENCA  
*desde 1867*

---



## Capítulo 3

# Diseño del Adaptador

En este capítulo se expone una idea general del funcionamiento del adaptador, y se detalla los procedimientos que se realizan en los diferentes módulos. En primer lugar, se presenta la interfaz para la comunicación entre el teléfono y el procesador, luego se enfoca en el funcionamiento del módulo Wi-Fi, analizando las diferentes funciones como: intercambio de paquetes UDP tanto para el establecimiento de la sesión como para el manejo de llamadas, y el intercambio de paquetes RTP para el transporte de la voz, y por último, el funcionamiento del MCU para el procesamiento de datos.

### 3.1. Diseño e implementación del circuito de interfaz

El circuito de interfaz consta de tres bloques, como se muestra en la Figura 3.1. El primero se encarga de recibir y enviar la señal de audio desde y hacia el teléfono. El segundo, se conecta en cascada al primero para adaptar las señales provenientes del mismo para que puedan ser vistas como de marcación y colgado, y el último es un timbre de aviso de llamada entrante.

El diseño del primer bloque se basa en una patente (vigente desde 2005) [12], la cual es un circuito electrónico diseñado con el fin de proveer una interfaz simple y económica que permita a los usuarios hacer uso de un teléfono común y corriente al realizar o recibir una llamada a través de Internet usando los puertos de micrófono y audífono de una computadora personal.

Un teléfono estándar combina un micrófono y un auricular conectados en serie. Debido a que la conexión normal utiliza únicamente dos cables, la señal eléctrica proveniente del teléfono es una mezcla de las señales del micrófono y el auricular. Este circuito se encarga de separar dichas señales desde la fuente para reducir el eco. Esta interfaz además trabaja de manera puramente analógica.

El bloque 2 consta de dos partes, una que envía la señal de voz proveniente del teléfono al MCU y la otra que transmite la voz hacia el teléfono. La Figura 3.2 muestra el circuito esquemático de la primera parte de este bloque. El teléfono tiene dos líneas de conexión, T1 y

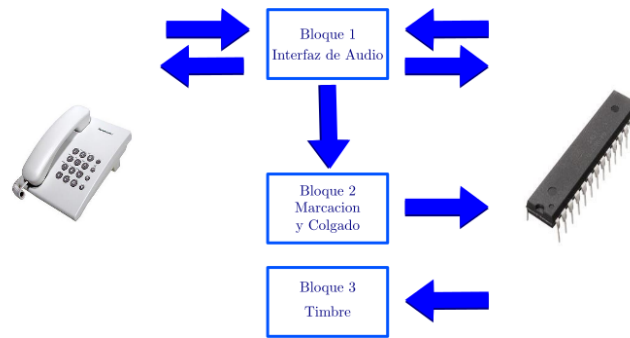


Figura 3.1: Diagrama de bloques del circuito de interfaz

T2. La línea T1 está conectada directamente a una fuente de poder de corriente directa, la cual sirve para alimentar el circuito interno del teléfono.

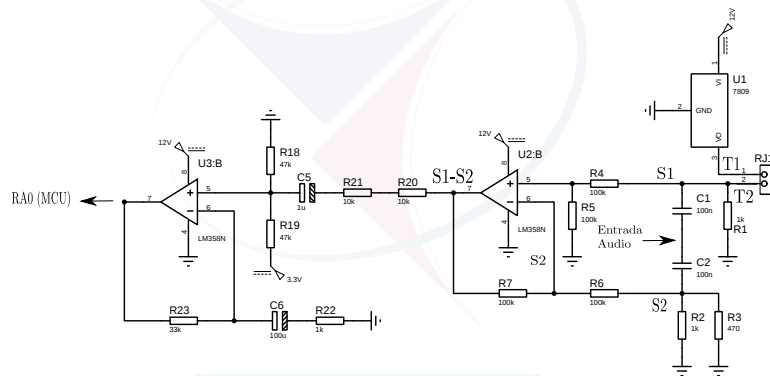


Figura 3.2: Circuito esquemático de la primera parte del bloque 1

La línea T2 se conecta a tierra mediante una resistencia, de donde se obtiene la señal S1 que contiene tanto la señal del micrófono como del auricular. Esta señal entra luego a un amplificador operacional configurado como restador. La señal de audio de entrada se conecta a la línea T2 mediante un capacitor. Esta señal de voz también pasa por otro capacitor para eliminar cualquier componente continua, S2, antes de entrar el amplificador operacional. El amplificador operacional se encarga de restar estas señales para así eliminar el eco producido por la conexión en serie del micrófono y el auricular. Luego la señal resultante entra a un preamplificador para que sea llevada a niveles de voltaje que puedan ser entendidos por el procesador.

La señal de audio de entrada proviene del DAC embebido en el microcontrolador y es acondicionada mediante el circuito mostrado en la Figura 3.3. Éste, además de ser utilizado para transmitir la señal de voz durante la llamada, también sirve para la señalización informativa, es decir, los tonos de invitación a marcar, ocupado y llamada. Cada país posee varios tonos informativos con diferentes frecuencias y tiempos en alto y bajo, las cuales se encuentran en la

Recomendación UIT-T E.180 [13]. Los tonos informativos para Ecuador tienen una frecuencia de  $425\text{Hz}$  y sus tiempos de cadencia se pueden ver en la Tabla 3.1.

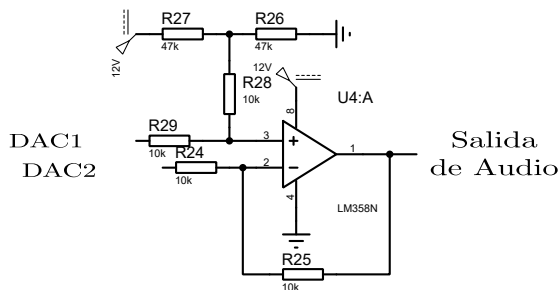


Figura 3.3: Circuito Esquemático de la segunda parte del bloque 1

Tono	Cadencia(s)
Tono de ocupado	0,33 on 0,33 Off
Tono de congestión	0,33 on 0,33 Off
Tono de invitación a marcar	continuo
Tono de llamada	1,2 on 4,65 Off
Tono de indicación de llamada en espera	0,2 on 0,6 Off

Tabla 3.1: Tiempos de cadencia utilizados en Ecuador

En la Figura 3.4 se observa el circuito esquemático del bloque 2. Mientras el teléfono está colgado, el voltaje en la línea  $T2$  será 0; al descolgar el teléfono, se cerrará su circuito interno, provocando una subida de tensión en la línea  $T2$ . Esta señal es comparada con un nivel de voltaje bajo, lo cual produce un estado bajo cuando el nivel en  $T2$  sea cero y un estado alto en caso contrario.

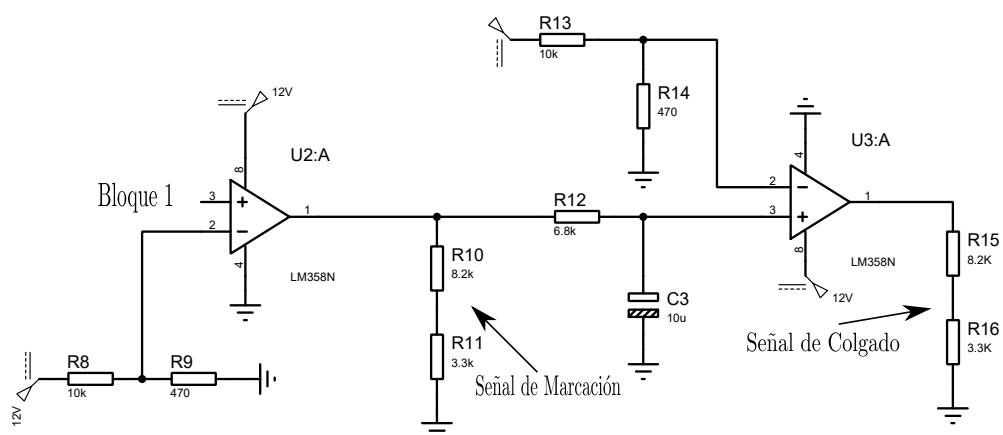


Figura 3.4: Circuito esquemático del bloque 2

Este primer comparador es utilizado para detectar la marcación, la cual se realiza mediante pulsos. Este método produce la conexión y desconexión del circuito de línea de manera muy rápida. De esta manera se generan pulsos de corriente directa que se emplean para indicar los dígitos marcados, tantos dígitos como pulsos se marquen, con excepción del cero que utiliza 10 pulsos. Los pulsos se generan a razón de 10 pulsos/seg. El tiempo entre dígitos es mayor para que cada secuencia pueda ser diferenciada [14] (ver Figura 3.5).

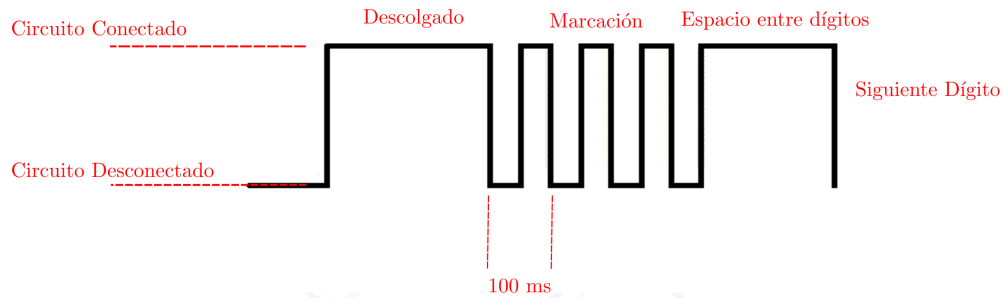


Figura 3.5: Marcación por pulsos

Debido al cambio de polaridad que se produce por la marcación de pulsos se utiliza un circuito RC con una constante  $\tau$  mayor a un semiperiodo del tren de pulsos, de esta manera el capacitor no se descargará completamente mientras el teléfono se encuentre descolgado. La tensión producida en el capacitor es comparada con un voltaje bajo para obtener un nivel de tensión alto o cero. Esta tensión se utiliza como señal de descolgado (*off-hook*). El circuito esquemático del bloque 3 se aprecia en la Figura 3.6. Para este adaptador se incluye un timbre externo debido a que la señal de timbre requerida para el teléfono es de  $90V_{ac}$  a  $20Hz$ , y debido a las limitaciones en costos del proyecto no se pudo implementar una solución económica para este problema, por lo que se optó por esta solución.

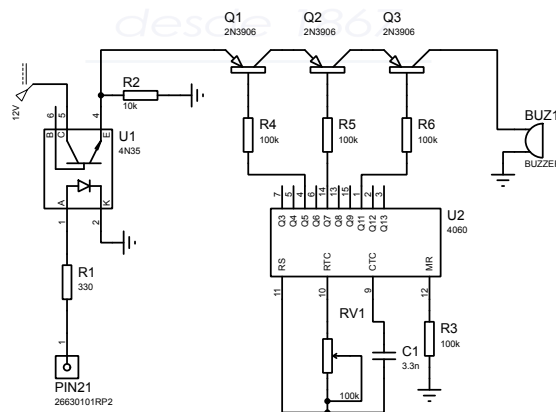


Figura 3.6: Circuito esquemático del timbre

El circuito generador del tono de timbre es muy simple y requiere una alimentación de únicamente 12V. El integrado CD4060B se utiliza para producir tres tipos de pulsos. Se configura

de tal manera que se obtenga pulsos de  $0,3125Hz$  en el pin 3. Al mismo tiempo, se obtienen pulsos de  $1,25Hz$  en el pin 1 y de  $20Hz$  en el pin 14. Estos tres pines están conectados a los terminales de base de los transistores. Utilizando un *buzzer*, se genera un tono de aproximadamente  $1kHz$ . En este circuito el *buzzer* es encendido y apagado a  $20Hz$  para el tono de timbre. Se obtienen pulsos de  $20Hz$  durante un periodo de  $400ms$ . Justo después de otro intervalo de  $400ms$ , los pulsos de  $20Hz$  vuelven a aparecer por otros  $400ms$ . Luego esto es acompañado por un intervalo de 2 segundos de silencio. Después el patrón se repite.

## 3.2. Configuración e implementación del módulo Wi-Fi

Para la comunicación entre el dispositivo [ATA](#) y la red, se hará uso del módulo inalámbrico ESP-8266 [15]. Es un chip integrado con conexión Wi-Fi compatible con el protocolo [TCP/IP](#), que soporta [IPv4](#) y los protocolos [TCP/UDP/HTTP/FTP](#). Este módulo puede ser configurado como punto de acceso, estación y una combinación de ambas. Dentro de este módulo se realizan diferentes procesos como: el establecimiento de conexión del adaptador a un punto de acceso, el establecimiento de la sesión con el servidor [VoIP](#) mediante el intercambio de paquetes [UDP](#) según el protocolo [SIP](#) presentado por la [IETF](#) en la [RFC-3261](#). Por último se establece un medio de transporte para el envío y recepción de paquetes [RTP](#).

### 3.2.1. Conexión del Adaptador a la Red

El dispositivo cuenta con un sistema de portabilidad, es decir que, el adaptador se podrá conectar a cualquier red existente. Esto se logra gracias a que los datos de la red pueden ser cambiados y almacenados en la memoria [EEPROM](#) del ESP-8266 mediante un servidor [WEB](#). Los diferentes pasos para conexión a la red se presentan en la Figura 3.7 y serán detallados más adelante.

- Al iniciar el adaptador por primera vez, este se inicializará como un servidor [WEB](#) simple. Un dispositivo cualquiera conectado a la red proporcionada por el ESP-8266, podrá hacer la petición [HTML](#) para la configuración del [Service Set Identifier \(SSID\)](#) y la contraseña de la red a la que se quiere conectar el módulo ESP-8266.
- Por el contrario, si el dispositivo ya tiene guardada información de una red en la [EEPROM](#), éste hace 3 intentos de conexión. Si dentro de estas 3 ocasiones no se logra la conexión, el dispositivo entrará en modo configuración y entonces se levanta el servidor [WEB](#) para que se vuelvan a introducir los datos de la red y pueda conectarse a la red. Además que cuenta con un botón con el cual se ingresa al modo configuración en cualquier momento.

### 3.2.2. Establecimiento de inicio de sesión

Una vez que el adaptador esté conectado a la red, procede a iniciar la sesión [SIP](#). Para esto, debe realizar el intercambio de mensajes con el servidor [VoIP](#). El adaptador envía el mensaje

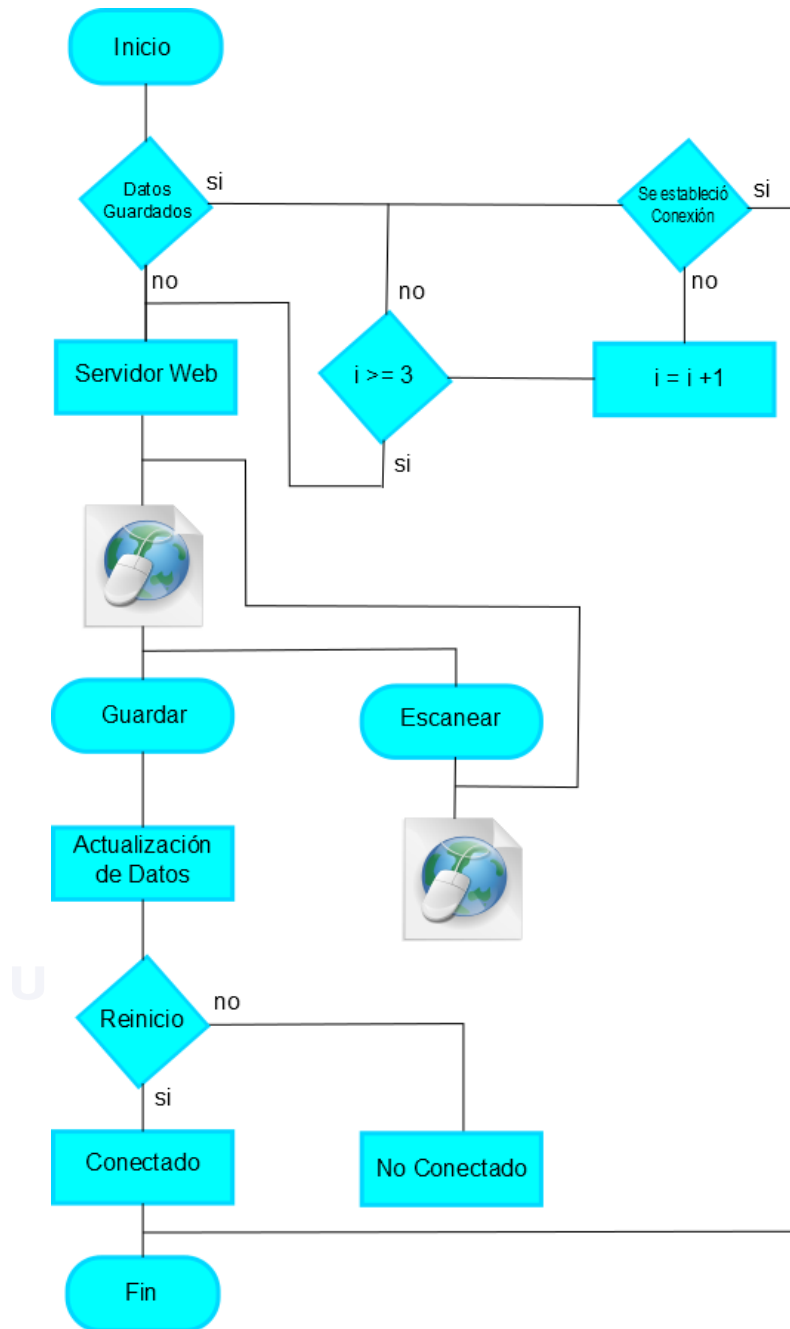


Figura 3.7: Flujograma para conexión a red

correspondiente al **REGISTER**, este mensaje consiste en diferentes campos que son necesarios para el establecimiento de la sesión como se muestra en la Figura 3.8.

El servidor responde con un mensaje **UNAUTHORIZED**, el cual quiere decir que no está auto-

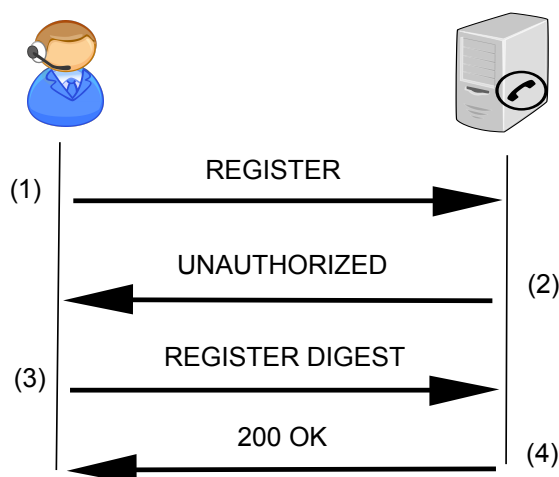


Figura 3.8: Formato básico de mensaje SIP

rizado para realizar esa petición. Por ende primero se autentica para poder realizar el inicio de sesión con éxito. Dentro del mensaje **UNAUTHORIZED** que nos envía el servidor se encuentran varios parámetros que sirven para la autenticación. Estos parámetros sirven para hacer el “Digest”, que es uno de los métodos de autenticación más usados en servidores para poder negociar credenciales, como el nombre del usuario y su contraseña. En este caso se utiliza una encriptación mediante *Message-Digest Algorithm 5 (MD5)*. Los parámetros que se utilizan para la obtención del código y la forma en la que se realiza se explican a continuación:

El agente de usuario SIP combina los parámetros “realm” y “nonce” junto con el nombre de usuario, contraseña, tipo de petición y la URL del servidor, teniendo la combinación siguiente:  $Str1 = MD5(username : realm : password)$ . La segunda combinación es:  $Str2 = MD5(método : urlservidor)$ . Donde método es el tipo de petición como: **REGISTER**, **INVITE** y **BYE**. La última combinación es:  $Response = MD5(str1 : nonce : str2)$ .

El agente de usuario envía nuevamente la petición **REGISTER** por segunda ocasión pero esta vez se le agrega una cabecera adicional la cual tiene los parámetros necesarios para autenticarse ante el servidor. El formato de esta cabecera es la siguiente:

- Authorization: Digest username=“001”, realm=“asteris”, nonce=“6b5bef1c”, response=“4bafec8d920b26740681dd286b316f85”, algorithm=md5

Si la encriptación realizada es correcta el servidor responde la petición con un mensaje **200 OK** el cual indica que el usuario se ha registrado exitosamente al servidor y puede realizar y recibir llamadas.

### 3.2.3. Establecimiento de invitación de llamada

- El proceso es similar al establecimiento de registro de usuario (ver Figura 3.9). Sin embargo, el mensaje **INVITE** presenta un cambio en el campo de longitud de contenido y un

aumento del campo “Content Type”. En este campo se establece el tipo de contenido en el cuerpo del mensaje, para este caso el contenido es de tipo aplicación *Session Description Protocol (SDP)* como se muestra en la Tabla 3.2. En este cuerpo de mensaje se proporciona principalmente el tipo de medio de comunicación (audio, video, texto, aplicación o mensaje). En este caso solo se utiliza el medio de comunicación de audio a través de *RTP* con una codificación *Pulse Code Modulation A-law (PCMA)/8000* (a-law G.711) que se comunica por el puerto 10100.

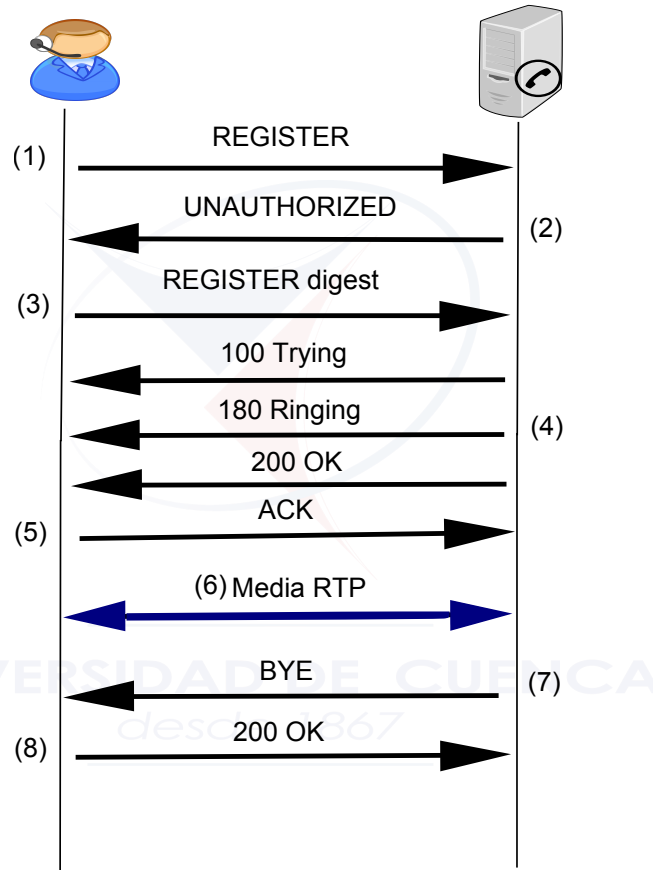


Figura 3.9: Formato básico de mensaje SIP

- Con el aumento del cuerpo del mensaje en la petición del **INVITE** se debe realizar un cálculo de la longitud del mensaje para agregarlo a la cabecera de longitud de contenido como se muestra en la Tabla 3.3.
- Cuando **Bob** realiza la petición de llamada a **Alice**, el teléfono suena pero **Alice** rechaza la llamada enviando un mensaje de estado final de error 486 BUSY indicando que ha rechazado la llamada como se muestra en la Figura 3.10b.
- Cuando **Bob** o **Alice** cancela una petición de **INVITE**, es decir, termina la llamada antes de que el otro conteste, lo hace mediante la petición **CANCEL** dando por terminado el



Parámetro <b>SDP</b>	Nombre del Parámetro
v=0	Número de Versión
o=00100 IN <b>IP</b> 4192,168,1,100	Identificador y host del usuario
c=IN <b>IP</b> 4192,168,1,101	Información de Conexión
t=00	Tiempo de sesión
m=audio 10100 <b>RTP/AVP</b> 0	Medio y Dirección de puerto
a=rtpmap:0 <b>PCMA</b> /8000	Atributos de la sesión

Tabla 3.2: Formato SDP

Línea	Total <i>Bytes</i>
v = 0	3
o = 00100 IN <b>IP</b> 4192,168,1,100	30
c = IN <b>IP</b> 4192,168,1,101	22
t = 00	5
m = audio 10100 <b>RTP/AVP</b> 0	23
a = rtpmap:0 <b>PCMA</b> /8000	20
	103

Tabla 3.3: Cálculo Longitud de Contenido

intento de llamada como se muestra en la Figura 3.10a.

#### 3.2.4. Media Session

En este punto se realiza el intercambio de los mensajes **RTP** que contienen en su carga útil, 160 *bytes*, que representan 20*ms* de audio hasta que uno de los dos dé por terminada la llamada.

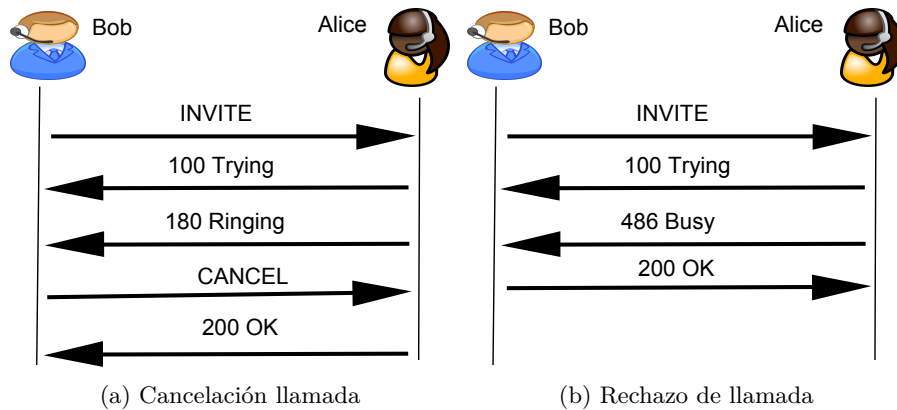


Figura 3.10: Flujos de llamada

### 3.3. Configuración y operación del microcontrolador

El microcontrolador está controlado por el teléfono y por el módulo Wi-Fi. Se encarga de verificar el estado del teléfono, tanto para proporcionar los tonos de marcación, llamada y ocupado como para comunicarle al módulo Wi-Fi cuando se desee realizar una llamada o finalizarla, interpreta los mensajes provenientes del módulo Wi-Fi para comunicar una llamada entrante o el estado de la llamada, y se encarga de procesar e intercambiar los mensajes de voz entre el teléfono y el módulo Wi-Fi.

Se utilizó el dsPIC33FJ128GP802, que es un PIC dedicado al tratamiento digital de señales, lo que lo hace ideal para el tratamiento de voz, además de que posee la interfaz de comunicación requerida. En la Figura 3.11 se puede apreciar la distribución de pines del microcontrolador. Los pines 13 y 28 se conectan a +3,3V, los pines 8, 27 y 29 se conectan a GND. El pin 20 se conecta a un capacitor cerámico de  $10\mu F$ , y en los pines 9 y 10 se conecta un cristal de  $8MHz$ .

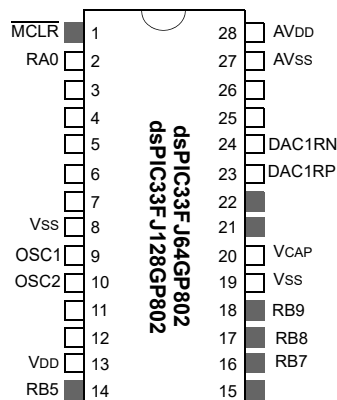


Figura 3.11: Distribución de pines del microcontrolador

Al pin **RB5** del microcontrolador se conecta la salida del comparador de voltaje para verificar el estado del teléfono. La detección de marcación se realiza mediante el pin **RB7**, aprovechando su función de interrupción externa para contar el número de pulsos. El pin **RB10** se conecta al circuito de timbre para activarlo cuando sea necesario. Los pines **RB8** y **RB9** han sido remapeados para ser utilizados en la comunicación serial con el módulo Wi-Fi. Los datos provenientes del módulo Wi-Fi se reciben en el pin **RB8**, mientras que en el pin **RB9** se envían los datos desde el microcontrolador al módulo Wi-Fi. El pin **RA0** se configura como una entrada analógica donde el controlador digitaliza la señal de voz proveniente del teléfono. El módulo **DAC** interno del microcontrolador (pines **DAC1RN** y **DAC1RP**) se usa para el envío de voz y los tonos de marcación, llamada y ocupado. Durante su funcionamiento, el dispositivo podría encontrarse en uno de los estados mostrados en la Figura 3.12 y que se explican a continuación.

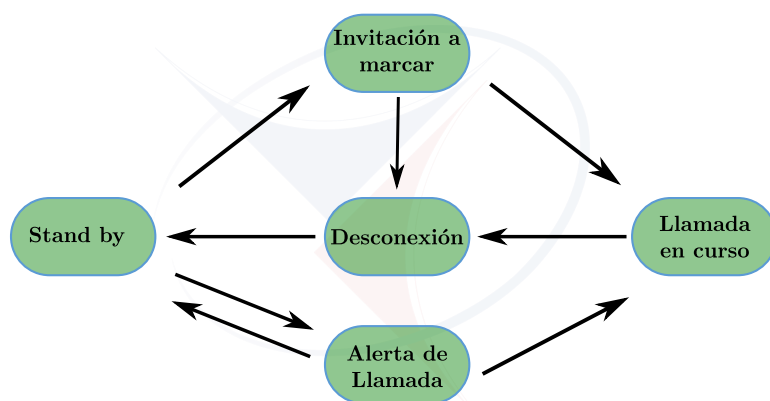


Figura 3.12: Diagrama de estados del dispositivo

**Stand by:** Es el estado inicial y final del dispositivo. El dispositivo permanece en este estado hasta que detecte una llamada entrante o hasta que el teléfono haya sido descolgado. Si se descuelga el teléfono para realizar una llamada, se pasa al estado de invitación a marcar y si el módulo Wi-Fi envía la señal de llamada entrante, se pasa al estado de alerta de llamada.

**Invitación a marcar:** Cuando el dispositivo entra en este estado, envía el tono de invitación a marcar hasta que la persona marque un número o el tiempo de espera expire. Al expirar el tiempo de espera se pasa al estado de desconexión, pero si se marca un número, el dispositivo tratará de realizar la llamada; lo cual puede llevar al dispositivo al estado de llamada en curso, si la llamada es exitosa, o al de desconexión en el caso contrario.

**Alerta de Llamada:** En este estado, el dispositivo hace sonar el timbre para informar de una llamada entrante. El timbre suena hasta que el teléfono sea contestado o el tiempo de espera expire. Si se contesta el teléfono, se pasa al estado de llamada en curso, mientras que si expira el tiempo de espera, se vuelve al estado de *Stand by*.

**Llamada en Curso:** Es en este estado en el que se realiza la codificación, envío, recepción

y decodificación de paquetes de voz. El dispositivo permanece en este estado hasta que una de las partes decida dar por finalizada la llamada, en cuyo caso se pasa al estado de desconexión.

**Desconexión:** En este estado, el dispositivo vuelve a su configuración inicial. También emite el tono de ocupado hasta que se cuelgue el teléfono. Luego de esto vuelve al estado de *Stand by*.

Durante el intercambio de voz por paquetes se produce retrasos provenientes de diferentes fuentes que pueden ser fijos (codificador, empaquetado y serialización) o variables (colas, conmutación y *jitter*). Para evitar estos problemas es necesario el uso de *buffers*.

La información es codificada y decodificada mediante **PCM**, utilizando un sistema de compansión. La compansión es sencillamente un sistema en el cual la información es comprimida, transmitida por un canal de banda ancha limitada y expandida en el lado del receptor. Es usado frecuentemente para reducir el ancho de banda requerido para transmitir voz por teléfono reduciendo las codewords de 13 *bits* a codewords de 8 *bits* [16].

Se hará uso del estándar A-law utilizado en Europa. En su forma digital, A-law es un sistema de compresión con pérdidas, lo que significa que, al recuperar la señal, esta no será igual a la original. Un código de entrada lineal de 13 *bits*, en formato signo-magnitud, es reducido a un código no linear de 8 *bits* formado como sigue: El *bit* más significativo es el *bit* de polaridad, los siguientes tres bits denotan el número de segmento y los últimos cuatro bits representan el número de paso de cuantización dentro del segmento [17]. La codificación y decodificación A-law se encuentran resumidas en las Tablas 3.4 y 3.5.

Segmento	Código de entrada lineal de 13 bits	Código comprimido de 8 bits
0	S0000000ABCDX	S000ABCD
±1	S0000001ABCDX	S001ABCD
±2	S000001ABCDXX	S010ABCD
±3	S00001ABCDXXX	S011ABCD
±4	S0001ABCDXXXX	S100ABCD
±5	S001ABCDXXXXX	S101ABCD
±6	S01ABCDXXXXXX	S110ABCD
±7	S1ABCDXXXXXXX	S111ABCD

Tabla 3.4: Codificación binaria A-law



Código comprimido de 8 bits	Código lineal de salida
S000ABCD	S0000000ABCD1
S001ABCD	S0000001ABCD1
S010ABCD	S000001ABCD10
S011ABCD	S00001ABCD100
S100ABCD	S0001ABCD1000
S101ABCD	S001ABCD10000
S110ABCD	S01ABCD100000
S111ABCD	S1ABCD1000000

Tabla 3.5: Decodificación binaria A-law



---

UNIVERSIDAD DE CUENCA  
*desde 1867*

---

## Capítulo 4

# Pruebas y resultados

En este capítulo se explican las distintas pruebas a las que fue sometido el dispositivo para verificar su funcionamiento. Lo que se desea principalmente es constatar que exista una correcta comunicación entre el dispositivo y el servidor, que tanto las señales informativas como las de marcación y estado de línea sean correctas y puedan ser entendidas, y que el intercambio de voz se realice sin problemas.

### 4.1. Señalización telefónica

Primero se verifica que las señales provenientes del teléfono hayan sido correctamente acondicionadas para poder ser tratadas por el microcontrolador. La Figura 4.1 muestra la señal de estado de línea, la cual es cero cuando el teléfono se encuentra colgado y mayor a cero en el caso contrario. La señal de marcación por pulsos obtenida se muestra en la Figura 4.2, como se puede ver esta señal solo tiene dos estados (conexión y desconexión), que era lo esperado.

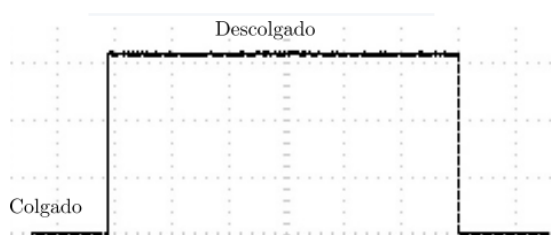


Figura 4.1: Señal de estado de línea

Luego se procede a comprobar que las señales informativas provenientes del microcontrolador sean correctas y estén acorde al estándar utilizado en Ecuador. Para disminuir el trabajo del procesador, en lugar de una señal sinusoidal se genera una onda cuadrada, la cual tiene una frecuencia bastante cercana a los  $425Hz$  que figuran en el estándar. Las señales de invitación a marcar, tono llamada y tono de ocupado se muestran en la Figura 4.3. Los tiempos de cadencia

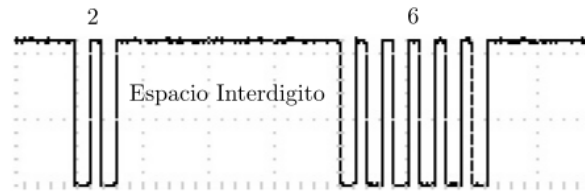


Figura 4.2: Señal de marcación por pulsos

también son bastante cercanos a los del estándar. La razón por la cual no se tiene la frecuencia o los tiempos de cadencia exactos es debido a que la frecuencia de trabajo del microcontrolador no es un múltiplo entero  $425Hz$ .

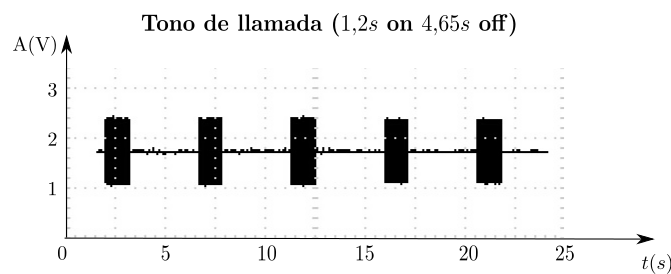
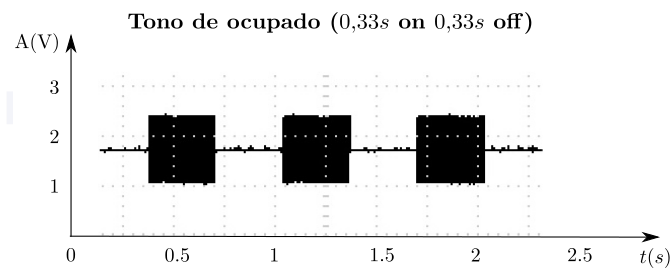
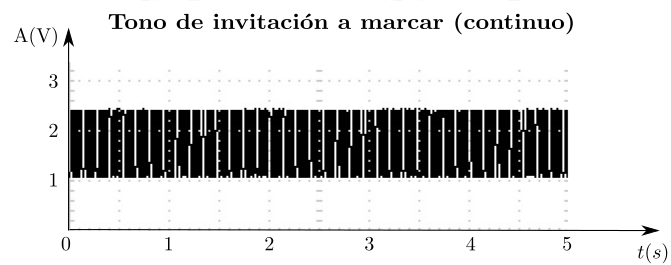


Figura 4.3: Señales informativas



## 4.2. Conexión con la red doméstica

Haciendo uso de la herramienta Serial Monitor del Software Arduino IDE y Wireshark, se puede realizar un *debug* del programa y de esta forma hacer un seguimiento del programa haciendo énfasis en las partes más importantes. Como se mencionó en la Sección 3.2.1 el módulo Wi-Fi se inicia como un AP. Lo que permite ingresar los datos de la red a la que se desea conectar mediante una página desarrollada sobre un servidor web. Para acceder a la página se usa cualquier navegador WEB y en la barra de direcciones se ingresa la dirección IP 192,168,4,1. Esta dirección es la puerta de enlace por defecto del módulo ESP-8266. Una vez ingresada esta dirección, el servidor web presenta la página que se muestra en la Figura 4.4.

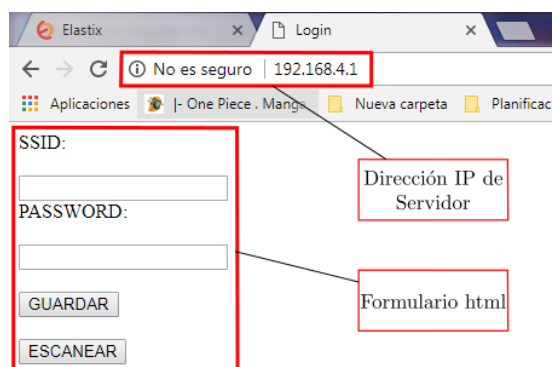
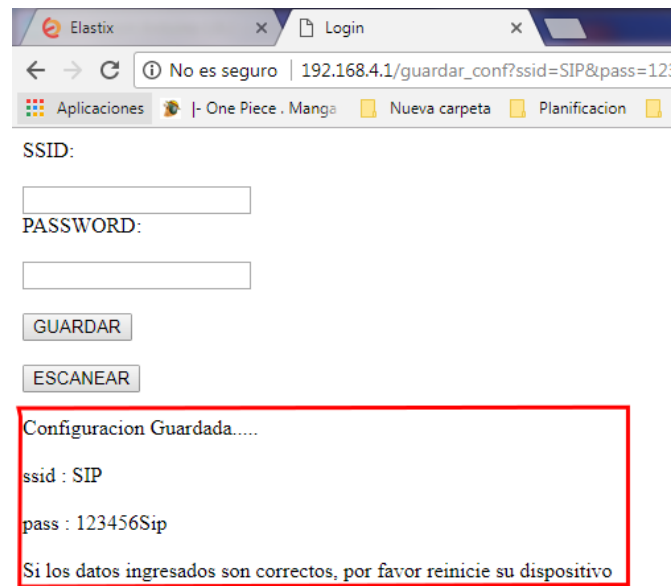


Figura 4.4: Página de Configuración

En esta página de configuración se puede observar dos objetos importantes: los botones y las cajas de texto. En las cajas de texto se ingresa el SSID y contraseña de la red doméstica a la que se desea conectar.

Y los dos botones: el primero “Guardar” este botón envía los datos ingresados en el campo anterior, al servidor para que este los guarde en la memoria EEPROM del dispositivo y el servidor retorna la página de configuración actualizada con el SSID y contraseña ingresados y un mensaje informativo como se muestra en la Figura 4.5. El segundo botón “Escanear”, solicita al servidor que realice un escaneo de todas la redes en el área y que retorne una página web con la información de todas las redes existentes como se muestra en la Figura 4.6.

Una vez almacenados los datos en la EEPROM se debe reiniciar el dispositivo para que proceda con el intento de conexión a la red cuyos datos fueron almacenados. Si los datos almacenados son correctos el dispositivo se conectará con éxito a la red y se mostrará un mensaje en el monitor serial con la SSID de la red y la dirección IP que le fue asignada como se muestra en la Figura 4.7.



SSID:

PASSWORD:

GUARDAR

ESCANEAR

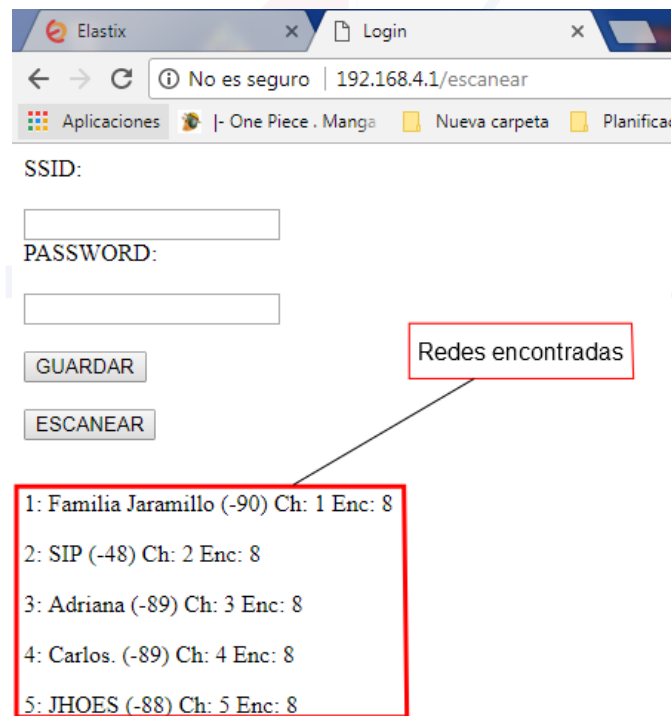
Configuracion Guardada.....

ssid : SIP

pass : 123456Sip

Si los datos ingresados son correctos, por favor reinicie su dispositivo

Figura 4.5: Botón Guardar



SSID:

PASSWORD:

GUARDAR

ESCANEAR

Redes encontradas

1: Familia Jaramillo (-90) Ch: 1 Enc: 8

2: SIP (-48) Ch: 2 Enc: 8

3: Adriana (-89) Ch: 3 Enc: 8

4: Carlos. (-89) Ch: 4 Enc: 8

5: JHOES (-88) Ch: 5 Enc: 8

Figura 4.6: Botón Escanear

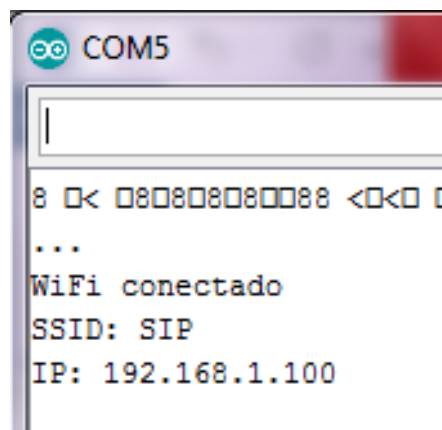


Figura 4.7: Conexión a la red

### 4.3. Conexión con el servidor VoIP

Para las pruebas de funcionamiento del dispositivo se instaló y configuró un servidor Elastix dentro de la misma red en la que se encuentra el dispositivo. En el servidor se configuraron tres extensiones UA. La extensión número 100 se le asignó al dispositivo y las otras dos extensiones la 200 y 201 fueron ocupadas por *softphone* para efectos de pruebas (Figura 4.8).

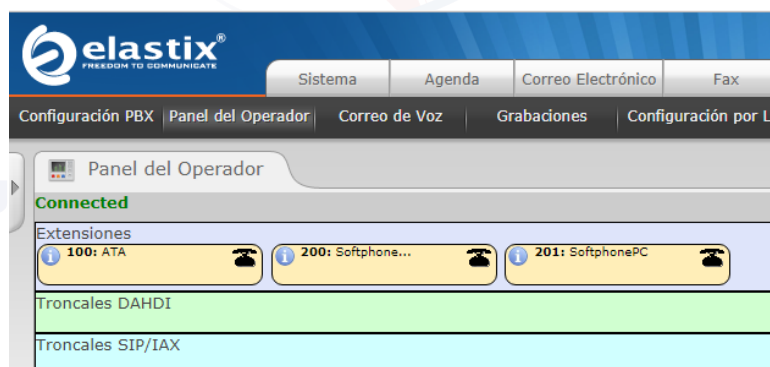


Figura 4.8: Servidor Elastix

Una vez instaladas las extensiones se hizo una prueba de registro del dispositivo al servidor VoIP. El módulo envía cada 5seg la petición de REGISTER sin autenticación hasta que el servidor responda la petición con un mensaje 401 UNAUTHORIZED. El cual como se mencionó en la Sección 3.2.1 contiene los parámetros necesarios para que el usuario realice la autenticación. Se envía por segunda ocasión la petición de REGISTER con los datos de autenticación y así se completa el registro del usuario en el servidor VoIP. En la Figura 4.9, se observa como en el servidor la extensión 100 cambia de un estado inactivo a un estado activo.

Se logró probar los diferentes escenarios de registro y llamada exitosamente.

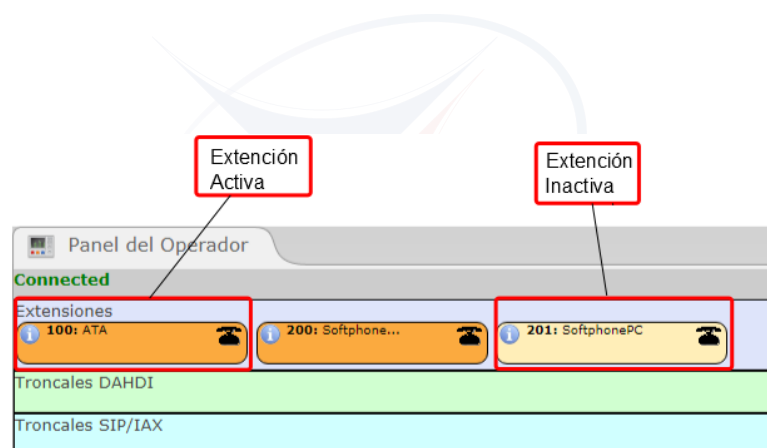


Figura 4.9: Estado de extensiones

## Capítulo 5

# Conclusiones y Recomendaciones

### 5.1. Conclusiones

Al culminar este proyecto, basados en los objetivos planteados, se llegó a las siguientes conclusiones.

El circuito de interfaz implementado, a pesar de ser muy sencillo, logra adaptar de forma correcta las señales provenientes del teléfono para su correcta interpretación en el microcontrolador. Además, logra reducir el eco lo suficiente como para que éste no sea un problema; aunque sigue presente, su nivel no crea inconvenientes durante la comunicación.

Se implementó el algoritmo de codificación A-law en su forma digital, primero realizando un muestreo y cuantificación lineal de la señal de voz, y luego comprimiendo este valor de 13 a 8 bits. La implementación de este algoritmo es de baja complejidad, por lo que su costo computacional es también bajo y prácticamente no introduce retardo algorítmico. Al comprimir el valor a 8 bits, se reduce el ancho de banda necesario para la transmisión de voz.

El sistema de marcación por pulsos resulta ser sencillo de implementar, sin embargo, su mayor problema es el tiempo que toma marcar un número y el no poder hacer uso de los símbolos especiales \* y #, además de que en muchos lugares este sistema se ha vuelto obsoleto. Por esta razón es conveniente cambiar al sistema de marcación por tonos (DTMF), lo cual no se realizó en este trabajo debido a inconvenientes surgidos durante su implementación.

Para el acceso a la red se analizó dos tecnologías de acceso, Ethernet y Wi-Fi. En base a esto, se optó por el módulo Wi-Fi, y de esta manera se proporcionó mayor portabilidad al dispositivo. Ya que con Ethernet se limitaba la movilidad del mismo. Además el módulo Ethernet que se estudió presentaba algunos problemas de compatibilidad con algunos modelos de enrutadores.

Se implementó un servidor web embebido en el dispositivo. Este presenta una interfaz para que la configuración del acceso a la red sea más amigable para el usuario.

Las cabeceras implementadas dentro de las peticiones [SIP request](#) utilizadas son suficientes para realizar un establecimiento de sesión para la transmisión de voz.



## 5.2. Recomendaciones

Se debe prestar atención a los tiempos de expiración de la sesión dentro de la petición de registro del usuario. Dentro de la [RFC-3565](#) se recomienda que transcurrido la mitad del tiempo expiración el usuario actualice su registro para evitar cualquier problema de desconexión con el servidor.

## 5.3. Trabajos Futuros

Aunque la interfaz implementada logra reducir el eco de buena manera, es importante que se estudie reemplazarla por una versión mejorada, ya sea con un circuito híbrido de telefonía o con una tarjeta de interfaz telefónica; ya que la presente interfaz causó inconvenientes al tratar de implementar un sistema de marcación por tonos.

Otra mejora importante que se puede realizar es el lograr que el dispositivo tenga un sistema de codificación adaptativo, ya que a pesar de que la codificación A-law es eficiente y fácil de implementar, es necesario contar con otros algoritmos; ya que, de esta manera, se puede cambiar entre codificaciones para escoger la que mejor se adapte a las condiciones de la red.

Las prestaciones del dispositivo desarrollado se limitan a realizar y recibir llamadas, por lo que es importante el aumentar los servicios que éste sea capaz de proporcionar, tales como un sistema de directorio telefónico o buzón de voz. También se debe aumentar la portabilidad del dispositivo para que el usuario pueda conectarlo con cualquier servidor de su preferencia.



# Anexos

---

UNIVERSIDAD DE CUENCA  
*desde 1867*

---





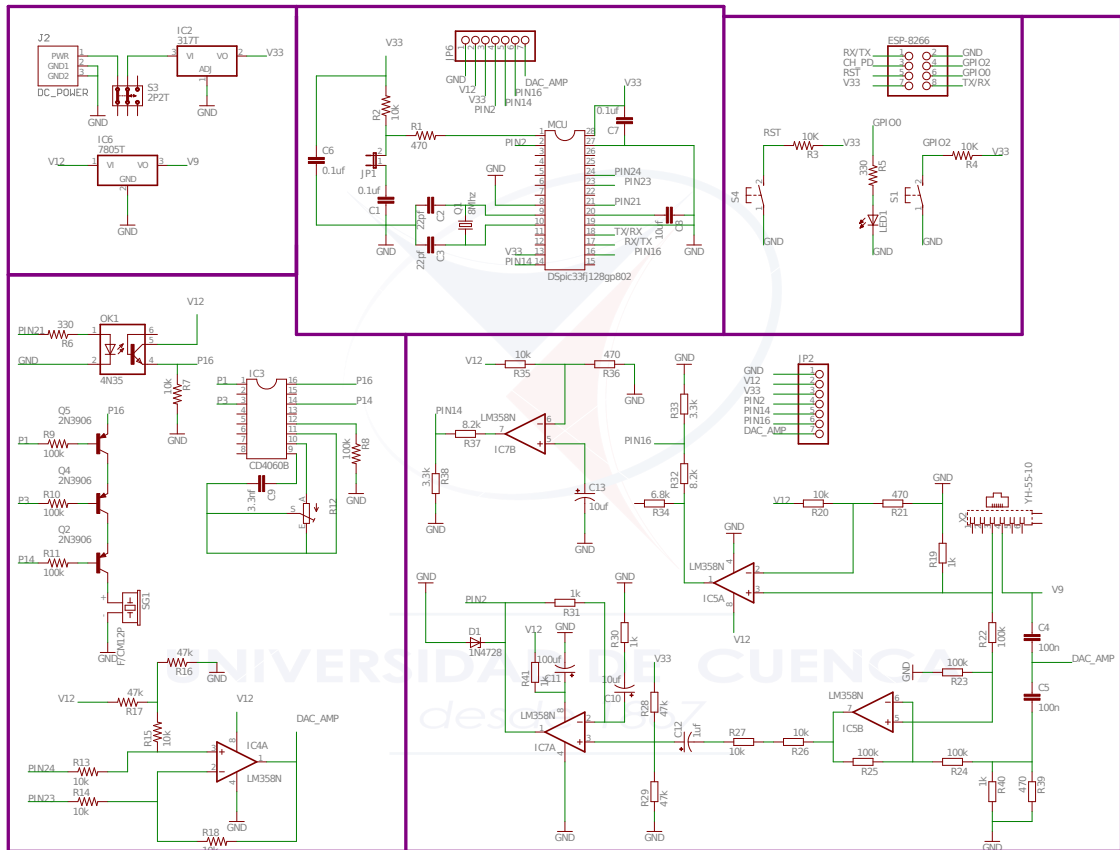


## Apéndice A

# Esquemático



UNIVERSIDAD DE CUENCA  
*desde 1867*



# Bibliografía

- [1] C. Abril y R. Arias, “Diseño e implementación de un prototipo nodo multimedia para servicios de voz sobre ip, navegación en internet, ubicacion e información de la ciudad de cuenca, para etapa ep,” Master’s thesis, Universidad de Cuenca, 2016.
- [2] D. Huidrobo, J y Róldan, *Tecnología VoIP y Telefonía IP*. Alfaomega, 2006.
- [3] A. Joaquín, *Voz IP (Servicios en red)*. Editex, 2011.
- [4] E. del Río, “Voz sobre IP: Fundamentos de la señalización mediante Protocolo SIP,” <http://telefonía.blog.tartanga.net/2014/04/25/voz-sobre-ip-fundamentos-de-la-senalizacion-mediante-el-protocolo-sip/>, Blog sobre Telefonía., 2014, online, visitado el 25-Julio-2017.
- [5] I. of Electrical y E. Engineers, “IEEE Standard for WiFi and WLAN applications,” IEEE, Standar, 2009.
- [6] Elastixtech, “Fundamentos de Telefonía VoIP,” <http://elastixtech.com/fundamentos-de-telefonía/voip-telefonía-ip/>, online, visitado el 25-Julio-2017.
- [7] F. Matango, “Antecedentes Históricos de Telefonía IP,” <http://www.servervoip.com/blog/antecedentes-historicos-de-telefonía-ip/>, 2006, online, visitado el 3-Julio-2017.
- [8] UIT, “Tiempo de transmisión en un sentido,” Unión Internacional de Telecomunicaciones, Recomendación, 2003.
- [9] M. Sulovic, D. Raca, M. Hadzialic, y N. Hadziahmetovic, “Dynamic Codec Selection Algorithm for VoIP,” in *ICDT 2011 : The Sixth International Conference on Digital Telecommunications*, 2011.
- [10] S. H., C. S., F. R., y J. V., “A Transport Protocol for Real-Time Applications,” IETF, Recomendación, 2003.
- [11] B. Johnston, *SIP: Understanding the Session Initiation Protocol*. Artech House, 2001.
- [12] M. Leshner y H. Leshner, “U.S. Patent No.6931120,” U.S. Patents, Tech. Rep., 2005.



- [13] UIT, “Diferentes tonos utilizados en las redes nacionales,” Unión Internacional de Telecomunicaciones, Recomendación, 2010.
- [14] E. Herrera, *Introducción a las telecomunicaciones modernas*. Editorial Limusa, 2004.
- [15] I. Grokhotkov, *ESP-8266 Arduino Core Documentation Release 2.4.0*, 2017.
- [16] Y. Engineering, “A-law/Mu-Law Companding,” <http://www.young-engineering.com>, online, visitado el 3-Julio-2017.
- [17] T. Ogunfunmi y N. Madihally, “Principles of Speech Coding. Taylor and Francis Group.” 2010.



---

UNIVERSIDAD DE CUENCA  
*desde 1867*

---